



TUGAS AKHIR - KS141501

**OPTIMASI PENJADWALAN WAKTU DAN RUANG
UJIAN OTOMATIS DENGAN MENGGUNAKAN
METODE *HYPER HEURISTIC* BERBASIS ALGORITMA
*GREAT DELUGE***

***AUTOMATED TIME AND ROOM EXAMINATION
TIMETABLING OPTIMIZATION USING HYPER
HEURISTIC BASED GREAT DELUGE ALGORITHM***

GUSTI BAGUS SYAHRANI
NRP 05211440000012

Dosen Pembimbing
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR - KS141501

**OPTIMASI PENJADWALAN WAKTU DAN RUANG
UJIAN OTOMATIS DENGAN MENGGUNAKAN
METODE *HYPER HEURISTIC* BERBASIS ALGORITMA
*GREAT DELUGE***

GUSTI BAGUS SYAHRANI
NRP 05211440000012

Dosen Pembimbing
Ahmad Muklason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT - KS141501

***AUTOMATED TIME AND ROOM EXAMINATION
TIMETABLING OPTIMIZATION USING HYPER
HEURISTIC BASED GREAT DELUGE ALGORITHM***

**GUSTI BAGUS SYAHRANI
NRP 05211440000012**

**Supervisors
Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

**INFORMATION SYSTEMS DEPARTMENT
Information and Communication Technology Faculty
Sepuluh Nopember Institut of Technology
Surabaya 2018**

LEMBAR PENGESAHAN

OPTIMASI PENJADWALAN WAKTU DAN RUANG UJIAN OTOMATIS DENGAN MENGGUNAKAN METODE *HYPER HEURISTIC* BERBASIS ALGORITMA *GREAT DELUGE*

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

GUSTI BAGUS SYAHRANI

NRP. 05211440000012

Surabaya, Juni 2018

**KETUA
DEPARTEMEN SISTEM INFORMASI**

Dr. Ir. Aris Triahyanto, M.Kom
NIP.19650310 1991021 001



LEMBAR PERSETUJUAN

OPTIMASI PENJADWALAN WAKTU DAN RUANG UJIAN OTOMATIS DENGAN MENGGUNAKAN METODE *HYPER HEURISTIC* BERBASIS ALGORITMA *GREAT DELUGE*

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

GUSTI BAGUS SYAHRANI

NRP. 05211440000012

Disetujui Tim Penguji : Tanggal Ujian : Juli 2018
Periode Wisuda : September 2018

Ahmad Muklason, S.Kom., M.Sc., Ph.D.

(Pembimbing I)

Edwin Riksakomara, S.Kom., M.T.

(Penguji I)

Faizal Mahananto, S.Kom, M.Eng., Ph.D.

(Penguji II)

OPTIMASI PENJADWALAN WAKTU DAN RUANG UJIAN OTOMATIS DENGAN MENGGUNAKAN METODE HYPER HEURISTIC BERBASIS ALGORITMA GREAT DELUGE

Nama Mahasiswa : Gusti Bagus Syahrani
NRP : 05211440000012
Departemen : Sistem Infromasi
**Dosen Pembimbing : Ahmad Muklason, S.Kom., M.Sc
Ph.D**

ABSTRAK

Permasalahan mengenai penjadwalan ujian di perguruan tinggi masih menjadi topik menarik untuk diteliti. Karena dalam ilmu komputer, optimasi penjadwalan ujian merupakan salah satu permasalahan NP-hard dimana belum ada algoritma eksak yang bisa menyelesaikannya dalam waktu polinomial. Selain itu pembuatan jadwal ujian dengan memastikan tidak ada satupun mahasiswa yang harus menempuh ujian dua mata kuliah dalam waktu yang sama, jumlah peserta ujian yang dijadwalkan tidak melebihi kapasitas ruang ujian, serta penentuan ruangan ujian bukanlah pekerjaan yang mudah, karena membutuhkan waktu yang cukup lama.

Pada Tugas akhir ini dihasilkan sebuah Aplikasi penjadwalan ujian berbasis website dengan menggunakan bahasa pemrograman PHP yang dapat menghasilkan jadwal ujian secara otomatis dengan menggunakan metode hyper heuristic berbasis great deluge.

Hasil dari tugas akhir in menunjukkan bahwa algoritma graph colouring - great deluge dapat digunakan untuk membuat

jadwal ujian yang lebih optimal dibandingkan dengan penjadwalan manual, penjadwalan menggunakan algoritma hill climbing dan algoritma simulated annealing dibuktikan dengan menghasilkan nilai proximity cost yang lebih rendah yaitu 29.41 sedangkan proximity cost yang dihasilkan dari penjadwalan ujian secara manual, menggunakan algoritma hill climbing dan algoritma simulated annealing berturut-turut adalah 51.15, 35.18 dan 35.84.

Kata kunci : penjadwalan ujian, penjadwalan otomatis, algoritma great deluge, metode hyper-heuristic.

***AUTOMATED TIME AND ROOM EXAMINATION
TIMETABLING OPTIMIZATION USING HYPER
HEURISTIC BASED GREAT DELUGE
ALGORITHM***

Name : Gusti Bagus Syahrani
NRP : 05211440000012
Departement : Sistem Infomasi
Supervisor : Ahmad Muklason, S.Kom., M.Sc
Ph.D

ABSTRACT

The problem of scheduling exam in college is still an interesting topic to be studied. Because in computer science, exam scheduling optimization is one of the NP-hard issues where there is no exact algorithm that can solve it in polynomial time. In addition, the preparation of the exam schedule to ensure that no student who has to take the exam of two courses in the same time and the number of test participants that scheduled are not exceed the capacity of the exam room, are not easy things to do, because it takes quite a long time.

In this final project generated an exam scheduling application website using PHP programming language that can generate test schedule automatically by using hyper heuristic method based on great deluge.

The results of this final project show that the graph colouring – great deluge algorithm can be used to schedule a more optimal tes schedule compared to manual scheduling, scheduling using hill climbing algorithm and simulated annealing algorithm is proved by producing lower of proximity cost of 29.41 while proximity cost which results from manual scheduling, using hill

climbing algorithm and simulated annealing algorithms respectively are 51.15, 35.18 and 35.84.

Keywords: exam scheduling, automatic exam scheduling, great deluge algorithm, hyper-heuristic method.

KATA PENGANTAR

Puji syukur kami panjatkan kehadiran Tuhan Yang Maha Esa, karena dengan limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan laporan tugas akhir yang berjudul :

“OPTIMASI PENJADWALAN WAKTU DAN RUANG UJIAN OTOMATIS DENGAN MENGGUNAKAN METODE HYPER HEURISTIC BERBASIS ALGORITMA GREAT DELUGE”

Tugas akhir ini merupakan salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terimakasih penulis sampaikan kepada pihak-pihak yang telah membantu, memberikan saran, semangat, motivasi dan mendukung baik dukungan moral maupun doa. Tugas akhir ini tidak akan terwujud tanpa dukungan dan bantuan dari berbagai pihak yang sudah meluangkan waktu, tenaga dan pikirannya. Atas berbagai bantuan tersebut, penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Allah SWT yang senantiasa melimpahkan berkah dan rahmat-Nya selama penulis mengerjakan Tugas Akhir.
2. Bapak Amirullah dan Ibu Qurrotul Ainiyah selaku orang tua dan wali penulis, terimakasih atas doa, bimbingan dan motivasi yang tak pernah henti diberikan kepada penulis.
3. Ade Irgi Normansyah, Faureza Arif Amrullah, dan Akifah Naila Risqi selaku adik dari penulis yang menjadi motivasi penulis untuk segera menyelesaikan tugas akhir ini, serta kepada sanak saudara dan keluarga besar yang telah mendukung dan mendoakan tiada henti untuk menyelesaikan Tugas Akhir ini.
4. Bapak Ahmad Mukhlason, S.Kom, M.Sc., Ph.D. selaku dosen pembimbing yang telah meluangkan waktu dan

- pikiran beliau untuk membimbing, memberikan ilmu, petunjuk dan motivasi selama pengerjaan Tugas Akhir.
5. Bapak Ir. Ahmad Holil Noor Ali, M.Kom selaku dosen wali yang selalu memberikan saran dan motivasi kepada penulis.
 6. Bapak Edwin Riksa Komara, S.Kom., M.T serta Bapak Faizal Mahananto, S.Kom., M.Eng., Ph. D. selaku dosen penguji yang telah memberikan saran ataupun kritik yang membangun dalam proses pengerjaan tugas akhir ini.
 7. Seluruh Dosen dan Karyawan Departemen Sistem Informasi yang telah memberikan ilmu yang bermanfaat dan membantu penulis selama menjalani masa perkuliahan di Departemen Sistem Informasi ITS.
 8. Seluruh narasumber dalam penelitian ini yang telah memberikan panduan dan masukan dalam pengerjaan Tugas Akhir ini.
 9. Kawan – kawan seperjuangan Redian Galih Irianti, Nur Achmad Setiadi, Muhammad Iqbal Imaduddin yang telah berjuang bersama, memberikan dukungan dan membantu dalam menyelesaikan Tugas Akhir.
 10. Keluarga besar kontrakan BME E-107 : Sultanto Hermawan Putra, Ilham Agung Aribowo, Rahardian Ahmad Fauzi, Azwin Rinaldy Fauzan, M. Rheza Kurniawan, Moh. Sulton Ali, Gusti Putra Pradana, Faizal Rahman dan Alwi Sina Khaiqiqi. Serta teman-teman seperantauan Satwika ITS 2014 yang selalu memberi dukungan dan bantuan, serta menjadi keluarga rantau penulis selama 4 tahun ini.
 11. Brilianto Wilis Satria Nugraha, Lutfia Nuzul Izzati, Redian Galih Irianti dan Riksa Rizki Zetta Adeli selaku sahabat penulis selama kuliah di Departemen Sistem Informasi yang telah memberikan bantuan, dukungan dan semangat kepada penulis.
 12. Silfia Rahmaawati, Desi Erliana Sari, Dewi Galuh Suciani yang telah membantu penulis dalam mengerjakan tugas akhir ini.

13. Keluarga Telkomsel *Flashlounge* ITS dan *Youth and Community* Telkomsel Surabaya Utara, khususnya Mbak Resieka yang telah memberikan dukungan, motivasi dan bantuan kepada penulis baik materi maupun moril.
14. Teman-teman Laboratorium RDIB, ADDI, dan MSI yang selalu menanyakan progress Tugas Akhir penulis.
15. Keluarga OSIRIS yang telah memberikan banyak pembelajaran, kenangan manis dan pahit semasa kuliah.
16. Dan seluruh pihak yang telah membantu penulis dalam mengerjakan tugas akhir ini yang tidak mungkin disebutkan satu per satu.

Surabaya, Juli 2018
Penulis,

(Gusti Bagus Syahrani)

DAFTAR ISI

LEMBAR PENGESAHAN.....	iii
LEMBAR PERSETUJUAN.....	iv
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
DAFTAR KODE.....	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah	2
1.3. Batasan Pengerjaan Tugas Akhir	3
1.4. Tujuan Tugas Akhir	3
1.5. Manfaat Tugas Akhir	4
1.6. Relevansi.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1. Penelitian Terdahulu	7
2.2. Dasar Teori	11
2.2.1. Penjadwalan Ujian	11
2.2.2. Algoritma Great Deluge	13
2.2.3. Algoritma Graph Colouring	15
2.2.4. Algoritma <i>Greedy</i>	17
2.2.5. Algoritma Hyper-Heuristic	18
2.2.6. Algoritma Knapsack Problem	19
BAB III METODOLOGI PENELITIAN.....	21
3.1. Metodologi Penelitian.....	21
3.1.1. Identifikasi Masalah	21
3.1.2. Penyiapan Data	21
3.1.3. Praproses Data.....	23
3.1.4. Pemodelan	24
3.1.5. Perancangan Algoritma.....	24
3.1.6. Uji Coba dan Analisis Performa Algoritma	24

3.1.7.	Analisis Jadwal Ujian.....	25
3.1.8.	Penyusunan Buku Tugas Akhir.....	25
BAB IV	PERANCANGAN	27
4.1.	Hasil Pengumpulan Data.....	27
4.1.1.	Data Peserta Ujian.....	27
4.1.2.	Data Mata Kuliah	28
4.1.3.	Data Ruang Ujian.....	30
4.2.	Pra-Proses Data.....	30
4.2.1.	Data CRS.....	31
4.2.2.	Data STU.....	33
4.2.3.	Data ROOM	34
4.3.	Formulasi Fungsi Tujuan	35
4.3.1.	Variabel Keputusan	35
4.3.2.	Batasan	36
4.3.3.	Fungsi Tujuan.....	37
4.3.4.	Matriks Konflik (C_{ij})	38
4.3.5.	Matriks Bobot Selisih Slot (W).....	38
4.4.	Implementasi Algoritma <i>Graph Colouring</i>	39
4.5.	Implementasi Algoritma <i>Great Deluge</i>	39
BAB V	IMPLEMENTASI	41
5.1.	Membaca Input File	41
5.1.1.	File CRS	41
5.1.2.	File STU	42
5.1.3.	File ROOM	42
5.2.	Pembuatan Matriks Konflik.....	43
5.3.	Penerapan Algoritma <i>Graph Colouring</i>	43
5.4.	Penerapan Algoritma <i>Great Deluge</i>	46
5.5.	Penerapan Algoritma Knapsack Problem	49
5.6.	Tampilan Aplikasi.....	50
5.6.1.	Halaman Utama.....	50
5.6.2.	Halaman Hasil Penjadwalan.....	51
5.6.3.	<i>Output File</i> Aplikasi.....	52
BAB VI	HASIL DAN PEMBAHASAN.....	55
6.1.	Data Uji Coba	55
6.2.	Lingkungan Uji Coba.....	55
6.3.	Fungsi Tujuan dari Jadwal Manual	56
6.4.	Hasil Uji Coba Initial Solution.....	57

6.5.	Hasil Uji Coba Algoritma <i>Great Deluge</i>	60
6.5.1.	Skenario Uji Coba 1 : Perubahan Iterasi	61
6.5.2.	Skenario Uji Coba 2 : Pemilihan <i>Low Level Heuristic</i>	67
6.5.3.	Skenario Uji Coba 3 : Perubahan Waktu	71
6.5.4.	Pembahasan Hasil Skenario Uji Coba.....	76
6.6.	Perbandingan dengan Algoritma Lain	76
6.7.	Penjadwalan Ujian Departemen Sistem Informasi .	78
6.7.1.	Algoritma <i>Great Deluge</i>	78
6.7.2.	Algoritma Simulated Annealing	79
6.7.3.	Perbandingan Algoritma <i>Great Deluge</i> dan Simulated Annealing.....	80
6.8.	Penjadwalan Ruang Ujian.....	82
6.8.1.	Penjadwalan Ruang Ujian Hasil Initial Solution	82
6.8.2.	Penjadwalan Ruang Ujian Hasil Algoritma <i>Great Deluge</i>	84
BAB VII SARAN DAN KESIMPULAN		87
7.1.	Kesimpulan	87
7.2.	Saran	87
DAFTAR PUSTAKA		89
BIODATA PENULIS		91
LAMPIRAN A : Jadwal UAS Manual.....		93
LAMPIRAN B : Inisiasi Jadwal UAS.....		99
LAMPIRAN C : Hasil Optimasi Jadwal UAS		105
LAMPIRAN D : Visualisasi Inisiasi Jadwal UAS.....		111
LAMPIRAN E : Visualisasi Hasil Optimasi Jadwal UAS...		113

DAFTAR GAMBAR

Gambar 1.1 Research roadmap laboratorium RDIB	5
Gambar 2.1 Pseudo-Code Algoritma Great Deluge Dalam Penjadwalan Ujian.....	14
Gambar 2.2 Graf tidak terarah $G=(V,E)$	15
Gambar 2.3 Model graf sederhana penjadwalan mata kuliah	16
Gambar 2.4 Framework Hyper-heuristic.....	19
Gambar 3.1 Metodologi Penelitian (Sumber : Penulis).....	22
Gambar 3.2 Contoh format carter dataset	23
Gambar 4.1 Potongan file .crs	33
Gambar 4.2 Potongan file .stu	34
Gambar 4.3 Potongan file .room	35
Gambar 5.1 Tampilan Halaman Utama Aplikasi	51
Gambar 5.2 Halaman Hasil Penjadwalan (1)	52
Gambar 5.3 Halaman Hasil Penjadwalan (2)	52
Gambar 5.4 <i>Output file</i> Aplikasi (1).....	53
Gambar 5.5 Output file Aplikasi (2)	53
Gambar 6.1 Permasalahan pada jadwal manual	56
Gambar 6.2 Hasil Penjadwalan Ujian Manual	57
Gambar 6.3 Grafik Proximity Cost Untuk Iterasi 1000	62
Gambar 6.4 Grafik Proximity Cost Untuk Iterasi 10000	63
Gambar 6.5 Grafik Proximity Cost Untuk Iterasi 100000	63
Gambar 6.6 Grafik Proximity Cost Untuk Iterasi 1000000 ...	64
Gambar 6.7 Grafik Box Plot Skenario 1	65
Gambar 6.8 Grafik Perbandingan Hasil Uji Coba Skenario 1	66
Gambar 6.9 Grafik Proximity Cost LLH Move	68
Gambar 6.10 Grafik Proximity Cost LLH Swap.....	69
Gambar 6.11 Grafik Proximity Cost LLH Move dan Swap...	69
Gambar 6.12 Grafik Box Plot Skenario 2	71
Gambar 6.13 Grafik Perbandingan Hasil Uji Coba Skenario 2	71
Gambar 6.14 Grafik Proximity Cost Dengan Waktu Eksekusi 1 Menit	72

Gambar 6.15 Grafik Proximity Cost Dengan Waktu Eksekusi 5 Menit73

Gambar 6.16 Grafik Proximity Cost Dengan Waktu Eksekusi 10 Menit74

Gambar 6.17 Grafik Box Plot Skenario 375

Gambar 6.18 Grafik Perbandingan Hasil Uji Coba Skenario 375

Gambar 6.19 Grafik Box Plot Perbandingan Algoritma Optimasi77

Gambar 6.20 Grafik Perbandingan Algoritma Optimasi78

Gambar 6.21 Grafik Proximity Cost Algoritma Great Deluge Pada Departemen Sistem Informasi79

Gambar 6.22 Grafik Proximity Cost Algoritma Simulated Annealing Pada Departemen Sistem Informasi.....80

Gambar 6.23 Grafik Perbandingan Proximity Algoritma Great Deluge dan Simulated Annealing81

Gambar 6.24 Grafik Box Plot Perbandingan Algoritma Great Deluge dan Simulated Annealing82

DAFTAR TABEL

Tabel 2.1 Penelitian terdahulu (1)	7
Tabel 2.2 Penelitian terdahulu (2)	8
Tabel 2.3 Penelitian terdahulu (3)	9
Tabel 2.4 Penelitian terdahulu (4)	10
Tabel 4.1 Ringkasan data	27
Tabel 4.2 Potongan data peserta ujian	27
Tabel 4.3 Data mata kuliah	28
Tabel 4.4 Data ruang ujian	30
Tabel 4.5 Konversi kode mata kuliah.....	31
Tabel 4.6 Potongan Hasil Pivot Tabel Data Peserta Ujian	33
Tabel 4.7 Matriks Konflik	38
Tabel 4.8 Matriks Selisih Bobot Slot	39
Tabel 6.1 Spesifikasi perangkat keras	55
Tabel 6.2 Spesifikasi perangkat lunak.....	55
Tabel 6.3 Jadwal bentrok	56
Tabel 6.4 Perbandingan Hasil Penjadwalan Manual, <i>Greedy</i> , dan <i>Graph Colouring</i>	58
Tabel 6.5 Hasil Penjadwalan Ujian Menggunakan Graph Colouring.....	60
Tabel 6.6 Parameter great deluge	60
Tabel 6.7 Perbandingan Hasil Uji Coba Skenario 1	65
Tabel 6.8 Hasil Jadwal Fungsi Tujuan Terbaik Pada Skenario 1	66
Tabel 6.9 perbandingan hasil percobaan skenario 2.....	70
Tabel 6.10 Perbandingan Hasil Percobaan Skenario 3.....	74
Tabel 6.11 Perbandingan Algoritma Optimasi	76
Tabel 6.12 Perbandingan Proximity Algoritma Great Deluge dan Simulated Annealing	80
Tabel 6.13 Potongan Tampilan Hasil Penjadwalan Ruang Pada Jadwal Initial Solution.....	83
Tabel 6.14 Hasil Penjadwalan Ruang Pada Jadwal Initial Solution	83
Tabel 6.15 Potongan Tampilan Hasil Penjadwalan Ruang Pada Jadwal Hasil Optimasi Great Deluge	85

Tabel 6.16 Hasil Penjadwalan Ruang Pada Jadwal Hasil Optimasi Great Deluge.....85

DAFTAR KODE

Kode 5.1 Membaca file	41
Kode 5.2 Membaca file crs	42
Kode 5.3. Membaca file stu	42
Kode 5.4. Membaca file room.....	43
Kode 5.5. Menghitung kapasitas total ruang	43
Kode 5.6. Membuat konflik matriks	43
Kode 5.7. Jadwal konflik.....	44
Kode 5.8. Method cekSlot.....	45
Kode 5.9. Method cekRoom	45
Kode 5.10. Method explore.....	46
Kode 5.11. Method Great Deluge	46
Kode 5.12. Pemilihan low level heuristic.....	47
Kode 5.13. Method move	47
Kode 5.14. Method swap.....	48
Kode 5.15. Great Deluge.....	49
Kode 5.16. Method bagiKelas.....	50

BAB I

PENDAHULUAN

Dalam bab ini akan dijelaskan gambaran umum mengenai tugas akhir yang diangkat. Hal tersebut meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir, dan relevan atau manfaat kegiatan tugas akhir. Selain itu, akan dijelaskan pula relevansi tugas akhir dengan laboratorium penelitian penulis.

1.1. Latar Belakang

Ujian merupakan salah satu alat pengukur mutu atau capaian yang krusial dalam sebuah proses pembelajaran. Dalam dunia perkuliahan, alat pengukur mutu atau capaian tersebut dapat dijadikan faktor penentu keberhasilan mahasiswa pada suatu mata kuliah. Hal tersebut menunjukkan pentingnya eksistensi ujian sebagai syarat baku kelulusan mahasiswa. Keberhasilan mahasiswa dalam ujian tidak hanya dipengaruhi oleh faktor internal dari pihak mahasiswa, namun juga dipengaruhi oleh faktor eksternal dari pihak kampus. Faktor eksternal tersebut seringkali kurang diperhatikan, padahal sebenarnya faktor ini memiliki peranan yang besar. Faktor eksternal yang harus disediakan pihak kampus salah satunya adalah *support system* yang optimal bagi mahasiswa.

Terwujudnya jadwal ujian yang memadai dilakukan melalui optimasi penjadwalan ujian dengan memperhatikan berbagai faktor yang mempengaruhinya. Hal tersebut diharapkan dapat berkorelasi positif dengan hasil capaian mahasiswa dalam ujian, salah satunya adalah mahasiswa menjadi memiliki waktu yang cukup untuk belajar sehingga dapat mempersiapkan ujian dengan lebih baik. Namun, pembuatan jadwal ujian yang optimal ini memiliki kendala. Hingga saat ini, pembuatan jadwal ujian masih dilakukan secara manual. Sehingga selain harus memastikan satu mahasiswa tidak memiliki lebih dari satu jadwal ujian dalam waktu yang bersamaan, pembuatan jadwal secara manual juga harus memastikan bahwa jumlah

peserta ujian yang dijadwalkan pada hari tersebut tidak melebihi kapasitas ruang ujian yang ada. Jumlah ujian yang dijalani mahasiswa juga menjadi pertimbangan karena semakin banyak jumlah ujian yang dijalani oleh satu mahasiswa dalam satu hari juga dapat berimbas nyata pada hasil ujian yang didapat. Dalam kasus lebih dari satu ujian pada hari yang sama, jeda antar ujian juga menjadi pertimbangan penting agar mahasiswa memiliki waktu cukup untuk belajar sebelum ujian selanjutnya dilaksanakan. Kompleksitas jadwal yang rumit dengan beberapa variable yang memiliki keterkaitan menjadikan proses pembuatan jadwal ujian bukanlah hal yang mudah dan membutuhkan waktu yang tidak sedikit.

Pada tugas akhir ini, metode hyper-heuristic berbasis pada algoritma great deluge akan digunakan untuk optimasi penjadwalan ujian, yang meliputi penjadwalan waktu dan ruang ujian. Metode hyper-heuristic dipilih karena merupakan metode baru yang lebih generic dibanding metode meta-heuristic. Sehingga judul yang diangkat dalam penelitian ini adalah “Optimasi Penjadwalan Waktu Dan Ruang Ujian Otomatis Dengan Menggunakan Metode *Hyper Heuristic* Berbasis Algoritma *Great Deluge*” dengan studi kasus di Departemen Sistem Informasi dan Departemen Teknik Industri. Dengan adanya penelitian ini, diharapkan optimasi penjadwalan dapat di implementasikan sehingga dapat memudahkan pihak manajemen kampus dalam pembuatan penjadwalan dan membawa manfaat bagi mahasiswa dalam mencapai hasil pembelajaran yang maksimal. Selain itu pada penelitian ini juga membandingkan dua algoritma yang berbeda yaitu algoritma *Great Deluge* dan algoritma pada penelitian sebelumnya[1] yaitu Simulated Annealing untuk menunjukkan algoritma mana yang lebih optimal.

1.2. Perumusan Masalah

Berdasarkan uraian latar belakang yang telah dijelaskan, maka rumusan masalah dari tugas akhir ini, yaitu :

1. Bagaimana membuat model matematis untuk permasalahan penjadwalan (waktu dan ruang) ujian di Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya?
2. Bagaimana menerapkan metode hyper-heuristic berbasis algoritma great deluge untuk menyelesaikan permasalahan penjadwalan (waktu dan ruang) ujian?
3. Bagaimana perbandingan hasil penjadwalan ujian menggunakan algoritma great deluge dengan algoritma simulated annealing serta penjadwalan secara manual?

1.3. Batasan Pengerjaan Tugas Akhir

Batasan permasalahan dalam tugas akhir ini, yaitu :

1. Data yang digunakan dalam tugas akhir ini yaitu data mahasiswa peserta dan data mata kuliah yang diujikan pada semester genap tahun 2016 – 2017 Departemen Sistem Informasi dan data mata kuliah yang diujikan pada UAS semester ganjil tahun 2017 – 2018 di Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya.
2. Hasil dari tugas akhir ini adalah sebuah aplikasi penjadwalan ujian otomatis yang dibangun dengan bahasa pemrograman PHP.
3. Tugas akhir ini berfokus pada penjadwalan berbasis waktu ujian dan lokasi ruang ujian.

1.4. Tujuan Tugas Akhir

Tujuan yang hendak dicapai dalam pengerjaan tugas akhir ini adalah:

1. Membuat model matematis untuk permasalahan penjadwalan ujian di Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.
2. Mengetahui hasil implementasi metode *hyper-heuristic* berbasis algoritma *great deluge* pada permasalahan

penjadwalan ujian Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

3. Mengetahui perbandingan hasil kinerja pendawalan ujian dengan menggunakan metode *hyper-heuristic* berbasis algoritma *great deluge* dengan algoritma *simulated annealing* serta secara manual.

1.5. Manfaat Tugas Akhir

Manfaat yang diberikan dengan adanya tugas akhir ini adalah sebagai berikut:

1. Manfaat Teoritis

- a. Dapat menambah khasanah keilmuan dalam bidang sistem informasi khususnya bidang optimasi penjadwalan ujian kuliah perguruan tinggi.
- b. Menambah referensi penelitian terkait implementasi metode *hyper-heuristic* berbasis algoritma *great deluge*.

2. Manfaat Praktis

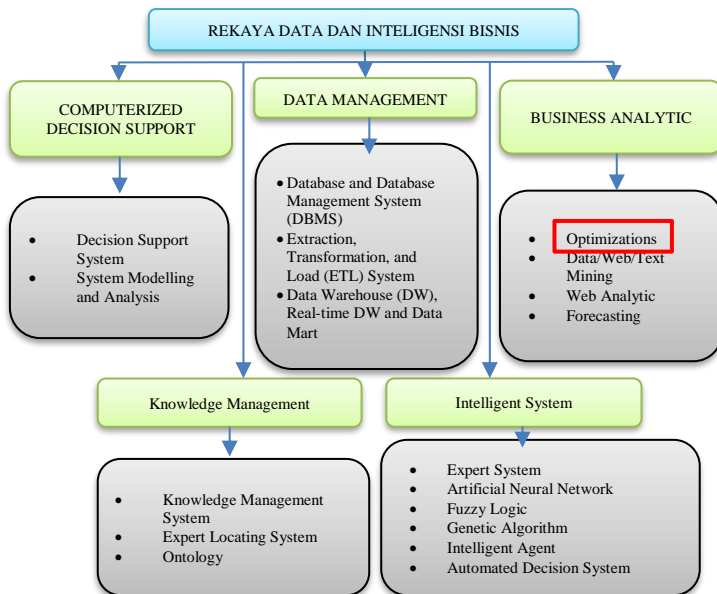
- a. Bagi Pihak Departemen Teknik Industri dan Departemen Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya
Tugas akhir ini diharapkan dapat membantu Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya untuk membuat jadwal ujian secara otomatis sehingga dapat menghemat waktu dan dapat menghasilkan jadwal ujian yang optimal bagi mahasiswa.
- b. Bagi Mahasiswa
Tugas akhir ini diharapkan dapat membantu mahasiswa memperoleh nilai yang maksimal dalam ujian. Mahasiswa dapat memperoleh jadwal ujian yang optimal, dalam satu hari mahasiswa diharapkan dapat mengikuti satu ujian saja sehingga lebih fokus belajar untuk mempersiapkan ujian tersebut.

c. Bagi Penelitian Berikutnya

Tugas akhir ini diharapkan dapat memberikan motivasi bagi peneliti lain untuk melakukan pengembangan terhadap studi mengenai penjadwalan ujian dengan menggunakan algoritma atau pendekatan lainnya.

1.6. Relevansi

Topik tugas akhir ini yaitu mengenai optimasi penjadwalan ujian yang memiliki relevansi dengan mata kuliah yang dipelajari sebelumnya yaitu Riset Operasi Lanjut. Selain itu, Topik tersebut sesuai dengan bidang ilmu riset operasi yang menjadi cakupan *research roadmap* pada laboratorium Rekayasa Data dan Inteligensi Bisnis (RDIB) yaitu Optimizations.



Gambar 1.1 Research roadmap laboratorium RDIB

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Sebelum melakukan pengerjaan tugas akhir, penulis melakukan tinjauan pustaka dari beberapa penelitian sebelumnya dan dasar teori yang sesuai dengan topik tugas akhir.

2.1. Penelitian Terdahulu

Pada sub bab ini akan diterangkan mengenai beberapa penelitian terdahulu yang telah dilakukan dan memiliki relevansi dengan tugas akhir ini. Penelitian terdahulu tersebut dapat dilihat pada tabel 2.1, 2.2, 2.3 dan 2.4.

Tabel 2.1 Penelitian terdahulu (1)

Nama Peneliti	Dian Kusumawardani, Ahmad Muklason [1]
Tahun Penelitian	2017
Judul Penelitian	<i>Optimasi Penjadwalan Ujian Otomatis Dengan Menggunakan Algoritma Greedy – Simulated Annealing – Hyper Heuristic (Studi Kasus : Departemen Sistem Informasi, ITS)</i>
Diskripsi Umum Penelitian	<p>Paper ini membahas tentang penjadwalan ujian otomatis di lembaga pendidikan, yaitu Departemen Sistem Infomasi, Institut Teknologi Sepuluh Nopember. Di dalam paper ini penulis menggunakan algoritma greedy – simulated annealing – hyper heuristic yang dibangun dengan menggunakan bahasa pemograman java.</p> <p>Penelitian ini melakukan uji coba algoritma simulated annealing dengan mengubah jumlah iterasi dan koefisien penurunan suhu, serta penentuan low level heuristic pada algoritma heuristic. Dan hasil dari uji coba tersebut menunjukkan bahwa iterasi yang mampu memberikan hasil fungsi tujuan terbaik yaitu 5000 iterasi, sedangkan untuk penentuan low</p>

	<p>level heuristic, metode swap yang sangat mempengaruhi hasil akhir dan fungsi tujuan. Untuk koefisien penurunan suhu, koefisien yang lebih kecil dapat menghasilkan jadwal dengan cepat, namun kurang optimal dibandingkan dengan koefisien yang lebih besar.</p> <p>Hasil dari penelitian ini menunjukkan bahwa algoritma greedy – simulated annealing – hyper heuristic lebih optimal dibandingkan dengan penjadwalan manual, dilihat dari nilai proximity cost yang lebih baik. Dimana jadwal yang dibuat secara manual menghasilkan proximity cost sebesar 39,570 , sedangkan hasil penelitian ini menghasilkan proximity cost sebesar 35,310.</p>
Keterkaitan Penelitian	<p>Pada tugas akhir ini melanjutkan penelitian dari paper ini, yaitu menggunakan metode hyper heuristic algoritma great deluge untuk penjadwalan otomatis dengan modifikasi algoritma dan dataset, disertai dengan pengalokasian ruang ujian.</p>

Tabel 2.2 Penelitian terdahulu (2)

Nama Peneliti	Ahmad Muklason [2]
Tahun Penelitian	2017
Judul Penelitian	<i>Solver Penjadwal Ujian Otomatis dengan Algoritma Maximal Clique dan Hyper-heuristic</i>
Deskripsi Umum Penelitian	Studi ini membahas tentang usulan metode baru mengenai permasalahan penjadwalan ujian yaitu metode heuristic sekuensial berdasarkan konsep maximal clique pada teori

	<p>graf yang digabung dengan metode hyper-heuristic.</p> <p>Hasil dari penelitian ini menunjukkan bahwa metode yang di usulkan sangat efektif dalam memecahkan permasalahan penjadwalan ujian di universitas atau institut, selain itu metode ini juga lebih unggul dibandingkan dengan penelitian sebelumnya yang menggunakan metode lain. Diharapkan penelitian selanjutnya dapat melakukan uji coba algoritma pada kasus yang nyata dilapangan, serta juga melibatkan penjadwalan ruang dan pengawas ujian.</p>
Keterkaitan Penelitian	<p>Pendekatan algoritma yang digunakan menjadi referensi untuk diterapkan dalam pengerjaan penelitian ini, yaitu menggunakan algoritma <i>great deluge</i> dan algoritma <i>graph coloring</i>.</p>

Tabel 2.3 Penelitian terdahulu (3)

Nama Peneliti	Ashis kumar Mandal, M N M Kahar [3]
Tahun Penelitian	2015
Judul Penelitian	<i>Solving Examination Timetabling Problem using Partial Exam Assignment with great Deluge Algorithm.</i>
Deskripsi Umum Penelitian	<p>Penelitian ini menyelesaikan permasalahan penjadwalan ujian menggunakan pendekatan partial exam assignment dengan algoritma great deluge. Dalam pendekatan ini, ujian di urutkan berdasarkan grafik dan hanya ujian yang dipilih yang dijadwalkan terlebih dahulu dan kemudian diperbaiki dengan menggunakan algoritma great deluge. Seluruh proses berlanjut sampai semua ujian telah dijadwalkan. Dalam penelitian ini penulis menggunakan dataset benchmark Toronto. Hasil percobaan menunjukkan bahwa algoritma great deluge dibandingkan dengan</p>

	pendekatan state-of-the-art menghasilkan solusi yang kompetitif untuk semua kasus.
Keterkaitan Penelitian	Pendekatan algoritma yang digunakan menjadi referensi untuk diterapkan dalam pengerjaan penelitian ini, yaitu menggunakan algoritma <i>graph coloruing</i> dan algoritma <i>great deluge</i> .

Tabel 2.4 Penelitian terdahulu (4)

Nama Peneliti	Salwani Abdullah [4]
Tahun Penelitian	2006
Judul Penelitian	<i>Heuristic Aproaches for University Timetabling Problems</i>
Penjelasan Singkat	Studi ini membahas tentang pendekatan algoritma heuristic dalam menyelesaikan penjadwalan di universitas, baik itu penjadwalan kuliah maupun penjadwalan ujian. Selain itu penulis juga memberikan pendekatan baru untuk pejadwalan ujian dengan mengimplementasikan large neighbourhood search. Penelitian ini menunjukkan bahwa metode tersebut sangat efektif dalam menghasilkan solusi berkualitas tinggi dengan mengorbankan waktu komputasi. Selain itu peneliti juga melakukan pendekatan multi-start, dimana dalam pendekatan tersebut menggabungkan antara dua algoritma heuristic yaitu great deluge dan simulated annealing. Hasil dari pendekatan multi-start ini hanya memperoleh satu hasil terbaik, namun hasil yang didapatkan layak dan berkualitas serta metode ini dapat mengurangi waktu yang dibutuhkan untuk memperoleh

	solusi terbaik dibandingkan dengan metode sebelum-sebelumnya.
Keterkaitan Penelitian	Algoritma yang digunakan dalam tugas akhir ini mengadopsi dari penelitian ini yaitu algoritma great deluge dengan metode heuristic, serta memberikan pemahaman mengenai penjadwalan ujian.

2.2. Dasar Teori

Pada sub bab ini akan dijabarkan mengenai dasar teori yang digunakan untuk mendukung pengerjaan tugas akhir ini.

2.2.1. Penjadwalan Ujian

Penjadwalan adalah suatu kegiatan mengalokasikan sumber daya yang terikat oleh kendala-kendala agar dapat ditempatkan pada ruang waktu, sehingga dapat memenuhi sedekat mungkin tujuan yang diinginkan [5]. Dalam definisi umum penjadwalan adalah mengalokasikan suatu peristiwa dengan sumber yang terbatas untuk memenuhi kendala (batasan) yang ditetapkan. Menurut [6], batasan dalam penyusunan jadwal ujian dibedakan menjadi dua yaitu hard constraint dan soft constraint. Hard constraint merupakan batasan yang harus terpenuhi untuk menyusun jadwal ujian, misalnya yaitu dalam satu waktu tidak diperbolehkan mahasiswa mengikuti dua ujian mata kuliah sekaligus. Sedangkan untuk soft constraint yaitu batasan yang tidak harus terpenuhi, namun sebisa mungkin dapat terpenuhi.

Masalah penjadwalan secara umum terdiri dari set peristiwa (E) seperti kuliah, ujian, dosen, laboratorium; set jumlah slot waktu (T) dan set batasan (C). Kendala umum yang sering terjadi pada penjadwalan adalah : (i) tidak ada 2 peristiwa yang dijadwalkan dalam satu waktu yang sama dan tempat yang sama. (ii) harus tersedia sumber daya yang memadai untuk semua peristiwa yang dijadwalkan untuk setiap timeslot[4]. Penjadwalan ujian merupakan penugasan ujian ke sejumlah periode waktu yang

tersedia sedemikian terbatas sehingga tidak ada konflik dan bentrokan (Carter dan Laporte, 1996)[7].

Penjadwalan ujian pada universitas merupakan suatu masalah yang kompleks, karena melibatkan banyak orang, dimana setiap universitas kebanyakan memiliki mahasiswa yang tidak sedikit. Oleh karena itu perlu dikembangkan sistem yang dapat mempermudah untuk melakukan penjadwalan ujian otomatis sehingga dapat digunakan di institusi Pendidikan seluruh dunia agar kualitas jadwal yang dihasilkan dapat memenuhi semua batasan yang ada.

Carter dkk. (1994)[8] mengemukakan bahwa tantangan mendasar dalam pembuatan jadwal ujian yaitu jumlah timeslot yang sedikit, sehingga dapat menimbulkan konflik dan untuk memenuhi beberapa constraint lainnya. Dimana konflik tersebut yang akan menjadi hard constraint sedangkan constraint lainnya sebagai soft constraint.

Burke dkk (1996)[9] melakukan survey mengenai kendala dalam pembuatan jadwal ujian di 56 universitas di dunia dan hasilnya sangat bervariasi dari universitas satu dengan universitas lainnya. Beberapa contoh hard constraintnya adalah :

- Ujian dengan jumlah mahasiswa terbanyak harus dijadwalkan lebih awal agar dapat memberi lebih banyak waktu untuk yang lainnya.
- Ujian tertentu harus dilakukan di ruangan tertentu, misalnya laboratorium.
- Jumlah kapasitas tempat duduk di ruangan tidak kurang dari jumlah mahasiswa yang ujian.
- Tidak ada mahasiswa yang harus mengikuti dua ujian secara bersamaan.
- Mata kuliah paralel dijadwalkan di waktu yang sama.

Sedangkan untuk soft constraintnya yaitu :

- Mahasiswa tidak dijadwalkan untuk mengikuti ujian lebih dari 1 kali dalam sehari.

- Mahasiswa tidak dijadwalkan mengikuti ujian dalam dua timeslot yang berurutan.

Dari beberapa pendapat diatas dapat disimpulkan bahwa tujuan utama dari penjadwalan ujian yaitu menjamin bahwa semua mahasiswa dapat melakukan ujian sesuai dengan kewajiban mata kuliah yang diambilnya. Namun dalam pembuatan jadwal juga harus memperhatikan beberapa sumber daya yang terkait yaitu ruangan, timeslot, dan lainnya. Untuk mencapai itu semua, beberapa constraint harus dipenuhi dan tingkat pemenuhan tersebut dapat menentukan kualitas jadwal yang dibuat, semakin banyak constraint yang terpenuhi maka semakin baik pula kualitas jadwal yang dibuat[4].

Menurut teori kompleksitas komputasi, permasalahan penjadwalan ujian diklasifikasikan sebagai *NP-complete problem*[10]. Dimana belum ada algoritma eksak yang dapat menemukan solusi optimal dalam waktu polinomial. Oleh sebab itulah, permasalahan ini masih menjadi permasalahan yang menarik untuk dikaji. *Benchmark dataset* untuk permasalahan penjadwalan ujian ini menggunakan *Dataset* yang umum digunakan adalah *Carter dataset* dan *International Timetabling Competition 2007 dataset*[11].

Pada tugas akhir ini, *dataset* yang akan digunakan adalah data peserta dan data mata kuliah yang diujikan pada ujian semester genap tahun ajaran 2016 – 2017 Departemen Sistem Informasi dan semester ganjil tahun ajaran 2017 – 2018 di Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya.

2.2.2. Algoritma Great Deluge

Great deluge merupakan algoritma metaheuristic local search yang dikembangkan oleh Dueck [3]. Algoritma ini merancang sebuah mekanisme untuk menghindari optima lokal dan berusaha memberikan pencarian solusi yang lebih luas [12] dengan menerima solusi terburuk yaitu apabila fungsi *penalty* berada dibawah nilai tertentu, perpindahan di lingkungan yang

terpilih akan diterima secara otomatis namun apabila diatas batas maka akan ditolak. Batas awal ditetapkan lebih tinggi dari biaya solusi awal dan akan dikurangi secara bertahap melalui proses perbaikan.

Algoritma ini membutuhkan dua parameter yaitu jumlah waktu komputasi yang diinginkan oleh pengguna dan perkiraan kualitas solusi yang dibutuhkan pengguna. Gambar 2.1 merupakan pseudo-code dari algoritma *great deluge* dalam penjadwalan ujian[4].

```

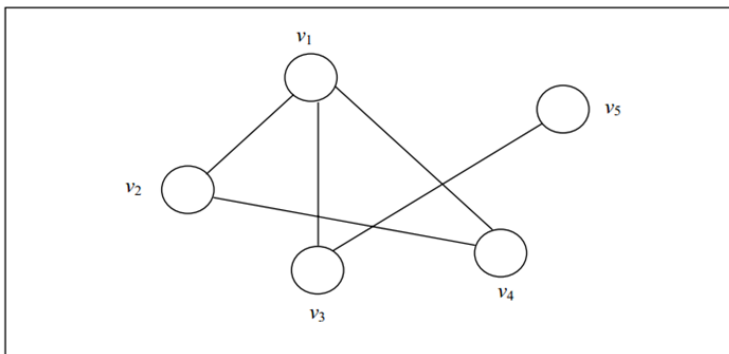
Set initial solution as  $Sol_{best}$  taken from large neighbourhood
search approach (Figure 6.1),  $Sol$ ;
Calculate the initial cost function value,  $f(Sol)$ ;
Set best solution,  $Sol_{best} \leftarrow Sol$ ;
Set estimated quality of final solution,  $estimatedquality$ ;
Set number of iterations,  $NumOfIte$ ;
Set initial level:  $level \leftarrow f(Sol)$ ;
Set decreasing rate  $\beta = ((f(Sol) - estimatedquality) / (NumOfIte))$ ;
Set iteration  $\leftarrow 0$ ;
Set not_improving_counter  $\leftarrow 0$ ;
do while (iteration <  $NumOfIte$ )
    Define neighbourhood of  $Sol$  by randomly assigning
    examination to a valid timeslot to generate a new solution
    called  $Sol^*$ ;
    Calculate  $f(Sol^*)$ ;
    if ( $f(Sol^*) < f(Sol_{best})$ )
         $Sol \leftarrow Sol^*$ ;
         $Sol_{best} \leftarrow Sol^*$ ;
        not_improving_counter  $\leftarrow 0$ ;
    else
        if ( $f(Sol^*) \leq level$ )
             $Sol \leftarrow Sol^*$ ;
            not_improving_counter  $\leftarrow 0$ ;
        else
            Increase not_improving_counter by 1;
            if (not_improving_counter == not_improving_
                length_GDA)
                exit;
            level = level -  $\beta$ ;
            Increase iteration by 1;
    end do;

```

Gambar 2.1 Pseudo-Code Algoritma Great Deluge Dalam Penjadwalan Ujian

2.2.3. Algoritma Graph Colouring

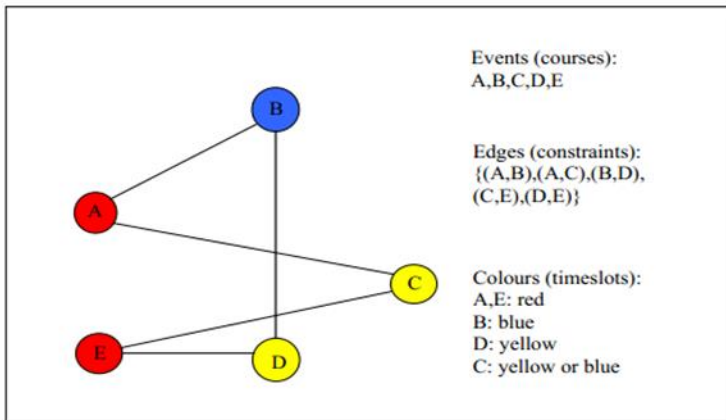
Graph colouring merupakan algoritma yang berkaitan dengan mewarnai simpul dari graf tertentu dengan menggunakan sejumlah warna tertentu. Dalam model graph colouring, simpul mewakili ujian, warna mewakili slot dan *edges* mewakili konflik antara ujian satu dengan ujian yang lainnya. Dimana setiap simpul grafik harus diwarnai dengan p sehingga tidak ada dua simpul yang menghubungkan dua *edge* dengan warna yang sama. Grafik di modelkan dengan $G = (V, E)$ dimana terdiri dari sekumpulan simpul $V = \{v_1, v_2, \dots, v_n\}$ dan sekumpulan dari *edge*[4].



Gambar 2.2 Graf tidak terarah $G=(V,E)$

Dari gambar di atas menunjukkan bahwa :

1. Derajat simpul adalah jumlah sisi yang menghubungkan dua simpul. Contohnya pada gambar X, vertex v_1 memiliki derajat 3, yaitu $(v_1, v_2), (v_1, v_3), (v_1, v_4)$.
2. Jumlah grafik bewarna merupakan jumlah minimum warna yang diperlukan untuk mewarnai sampul, sehingga tidak ada 2 simpul yang dihubungkan oleh *edge* keduanya memiliki warna yang sama.



Gambar 2.3 Model graf sederhana penjadwalan mata kuliah

Dari gambar 3, kita dapat melihat bahwa terdapat lima mata kuliah yaitu A, B, C, D dan E. salah satu tujuan yang mungkin yaitu mencari minimum jumlah timeslot yang dibutuhkan untuk menjadwalkan lima mata kuliah. Garis yang menunjukkan antara mata kuliah satu dengan yang lainnya menunjukkan bahwa dua matakuliah tersebut terdapat bentrok sehingga kedua mata kuliah tersebut tidak dapat dijadwalkan bersamaan. Sebagai contoh dari gambar 3 yaitu mata kuliah B tidak bisa dijadwalkan bersamaan dengan mata kuliah A dan D. dari gambar diatas menunjukkan bahwa terdapat 3 warna dalam pewarnaan simpul yaitu biru, kuning dan merah. Dimana jumlah warna tersebut, menunjukkan jumlah timeslot[4].

Terdapat banyak teknik pengurutan heuristik berurutan dalam menghasilkan solusi awal. Grafik pendekatan yang banyak digunakan adalah sebagai berikut [3] :

- Largest degree (LD) : teknik ini mengurutkan ujian berdasarkan jumlah konflik terbesar ujian.
- Largest weighted degree (LWD) : heuristik ini mengurutkan ujian berdasarkan jumlah siswa yang berkonflik yaitu jadwal ujian dengan jumlah siswa yang terlibat dalam konflik.

- Largest enrollment (LE) : ujian dengan pendaftar ujian terbanyak dijadwalkan terlebih dahulu.
- Saturation degree (SD) : heuristik ini mengurutkan ujian berdasarkan jumlah sisa waktu yang tersisa, ujian dengan periode sedikit diberikan prioritas yang lebih tinggi untuk dijadwalkan terlebih dahulu.

2.2.4. Algoritma *Greedy*

Algoritma *greedy* merupakan algoritma populer yang sering digunakan dalam menyelesaikan permasalahan optimasi dengan mencari solusi paling optimal. Prinsip kerja dari algoritma *greedy* yaitu dengan memilih langkah solusi yang terbaik tanpa memperhatikan langkah solusi selanjutnya. Sehingga dalam setiap langkahnya dipilih keputusan yang terbaik pada saat itu saja (*local optimum*) dengan harapan setiap langkah tersebut dapat mengarahkan kepada solusi *global optimum*[13]. Pada algoritma *greedy* terdapat 5 komponen yang perlu diperhatikan, antara lain[14] :

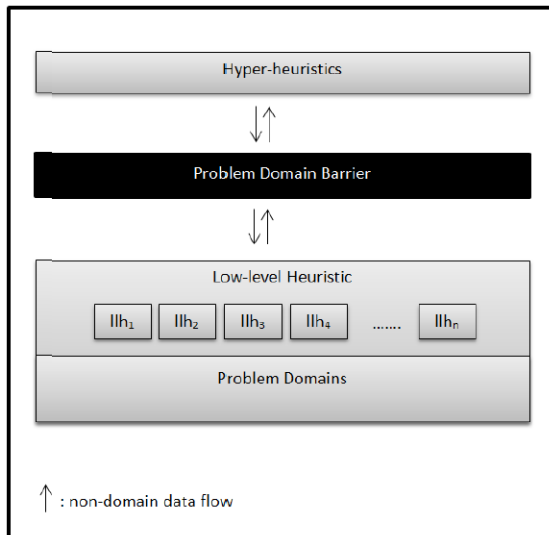
1. Himpunan kandidat (C)
Himpunan kandidat merupakan kumpulan kandidat elemen yang nantinya akan membentuk solusi.
2. Himpunan solusi (S)
Himpunan solusi merupakan kumpulan dari elemen-elemen yang merupakan solusi terbaik dalam setiap langkah.
3. Fungsi seleksi
Fungsi seleksi merupakan fungsi untuk memilih kandidat terbaik dalam setiap langkahnya untuk dimasukkan ke dalam solusi untuk mendapatkan optimum global.
4. Fungsi kelayakan
Fungsi kelayakan merupakan fungsi yang digunakan untuk menentukan apakah sebuah kandidat layak dimasukkan ke dalam solusi.
5. Fungsi tujuan
Fungsi tujuan bertujuan untuk memberikan penilaian terhadap solusi.

Kelemahan dari algoritma *greedy* yaitu solusi optimal global belum tentu menjadi solusi optimum yang terbaik, namun sub-optimum atau *pseudo-optimum*. Hal ini dikarenakan algoritma *greedy* tidak melakukan pengecekan menyeluruh terhadap solusi yang ada. Oleh karena itu, penggunaan algoritma perlu digabungkan dengan algoritma yang lain untuk mendapatkan solusi yang lebih optimal.

2.2.5. Algoritma Hyper-Heuristic

Hyper-heuristic adalah proses menggunakan meta-heuristic untuk memilih (meta) heuristik untuk memecahkan masalah (Burke dkk, 2003c). Secara umum *hyper-heuristic* adalah sebuah kumpulan pendekatan atau metode yang memiliki tujuan untuk otomatisasi suatu proses dengan menggunakan machine learning untuk memilih dan mengkombinasikan heuristik yang lebih sederhana atau menghasilkan heuristik baru dengan komponen heuristik yang sudah ada, sehingga menghasilkan heuristik yang bersifat general yang dapat digunakan untuk menyelesaikan kumpulan permasalahan yang berbeda[11]. *Hyper-heuristic* merupakan heuristik tingkat tinggi yang dapat memodifikasi solusinya secara tidak langsung dengan menggunakan beberapa heuristik tingkat rendah berdasarkan mekanisme pembelajaran[4].

Gambar 2.5 menunjukkan framework dari *hyper-heuristic* [15]. Pada framework ini algoritma hyper-heuristic menggunakan *problem domain barrier* untuk menghubungkan antara *low level heuristic* dan *hyper-heuristic*, sehingga *hyper-heuristic* tidak bersinggungan langsung dengan solution space.



Gambar 2.4 Framework Hyper-heuristic

Dengan adanya *search space* yang lebih tinggi ini, algoritma *hyper-heuristic* tidak bergantung pada *problem domain* tertentu saja sehingga tidak diperlukan *parameter tuning* secara manual untuk setiap *problem domain*. Hal ini dikarenakan algoritma *hyper-heuristic* sudah mampu melakukan mekanisme *parameter tuning* secara otomatis [2].

2.2.6. Algoritma Knapsack Problem

Knapsack problem merupakan permasalahan optimasi kombinatorial dimana terdapat n item yang mungkin dapat dimasukkan ke dalam karung. Setiap item i memiliki bobot w_i dan *value* v_i . dimana $w_i \geq 0$ dan $v_i \geq 0$. Terdapat batas W pada berat total item yang dapat dimasukkan ke dalam karung. Tujuannya adalah untuk memilih item yang dapat dimasukkan ke dalam karung dengan nilai total yang maksimal [16].

Knapsack problem dapat di rumuskan sebagai *zero-one programming problem* dengan variabel keputusan (1) :

$$x_i = \begin{cases} 1, & \text{jika item } i \text{ dimasukkan dalam karung} \\ 0, & \text{jika tidak} \end{cases} \quad (1)$$

dengan fungsi tujuan (2) yaitu :

$$\text{Maximize } \sum_{i=1}^n v_i x_i \quad (2)$$

Dan batasan (3) :

$$\sum_{i=1}^n w_i x_i \leq W \quad (3)$$

Dimana,
 $i = 1, 2, 3, \dots, n$
 $x_i = \{0, 1\}$

BAB III METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan mengenai alur metodologi yang akan dilakukan dalam tugas akhir ini. Metodologi ini juga digunakan sebagai pedoman untuk melaksanakan tugas akhir agar terarah dan sistematis. Gambar 3.1 menunjukkan metodologi tugas akhir yang digunakan.

3.1. Metodologi Penelitian

Diagram Metodologi dari Tugas Akhir ini dapat dilihat pada gambar 3.1.

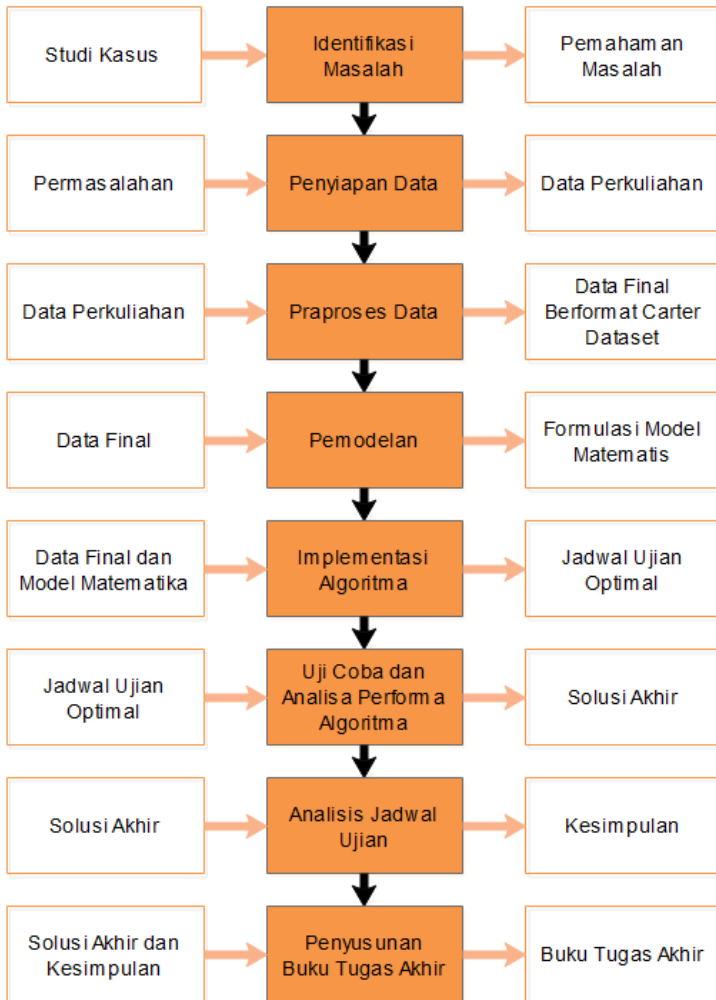
3.1.1. Identifikasi Masalah

Tahap identifikasi masalah merupakan tahap untuk memahami permasalahan di Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya mengenai penjadwalan ujian dimana penjadwalan ujian masih dilakukan secara manual dan masih sering diperlukan banyak revisi untuk mendapatkan jadwal yang tepat, selain itu pada tahap ini juga menentukan tujuan dan batas penelitian tugas akhir yang dikerjakan. Serta studi literatur mengenai penelitian-penelitian sebelumnya dan pendekatan-pendekatan yang diperlukan untuk memecahkan permasalahan penjadwalan ujian.

3.1.2. Penyiapan Data

Pada tahap penyiapan data, dilakukan pengumpulan data-data awal yang diperlukan dalam pengerjaan tugas akhir yaitu dengan melakukan wawancara kepada Effi Latiffianti, ST., M.Sc selaku Sekretaris Program Studi Sarjana (S1) Departemen Teknik Industri. Data yang didapatkan dari hasil wawancara adalah data peserta, data mata kuliah dan data ruang ujian yang diujikan pada Ujian Akhir Semester (UAS) pada semester ganjil tahun 2017/2018 di Departemen Teknik Industri Institut Teknologi Sepuluh Nopember Surabaya program studi

Sarjana (S1) dan data UTS dan UAS pada semester genap 2016/2017 Departemen Sistem Informasi Institut Teknologi Sepuluh Nopember Surabaya program studi Sarjana (S1) yang didapatkan dari penelitian sebelumnya [1].



Gambar 3.1 Metodologi Penelitian (Sumber : Penulis)

3.1.3. Praproses Data

Tahap ini merupakan tahap untuk memahami data yang diperoleh, yaitu memahami tipe datanya, variabel-variabel yang melekat pada data, dan jumlah data. Tahap selanjutnya adalah mempersiapkan data final yang akan digunakan untuk proses pemodelan. Data final yang dimaksud adalah data yang telah melalui tahap ini sehingga menjadi data yang berkualitas baik dan dapat diolah. Format data yang digunakan akan disesuaikan dengan *Carter Dataset*, dimana data yang semula berbentuk file *excel* akan diubah menjadi file dengan ekstensi *.txt*. Keluaran dari tahap ini adalah sebuah data final berformat *Carter Dataset* yang akan digunakan pada tahap berikutnya. Gambar 6 merupakan contoh dari *format carter dataset*, (atas) merupakan carter dataset dengan format *.crs* yang menunjukkan mata kuliah dengan jumlah mahasiswa yang mengambilnya, sedangkan gambar 6 (kiri) merupakan carter dataset dengan format *.stu* yang menunjukkan mata kuliah kuliah yang diambil mahasiswa i.

1	0001	4
2	0002	155
3	0003	150
4	0004	154
5	0005	152
6	0006	64
7	0007	157
8	0008	17

558	0001	0002	0003	0004	0005
559	0002	0003	0004	0005	0009
560	0002	0003	0004	0005	
561	0002	0003	0004	0005	
562	0002	0003	0004	0005	0006
563	0002	0003	0004	0005	0006
564	0002	0003	0004	0005	
565	0002	0003	0004	0005	0006
566	0002	0003	0004	0005	0006

Gambar 3.2 Contoh format carter dataset

3.1.4. Pemodelan

Tahap pemodelan adalah tahap pembuatan model matematis dari permasalahan penjadwalan ujian yang ada di Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Model yang dibuat harus memperhatikan beberapa variabel seperti : variabel keputusan, fungsi tujuan, dan batasan. Setelah model dibuat, maka dapat dilanjutkan ke tahap perancangan algoritma.

3.1.5. Perancangan Algoritma

Pada tahap perancangan algoritma dilakukan penerapan algoritma *graph colouring*, *great deluge* berbasis *hyper heuristic* untuk penentuan timeslot yang optimal. Selanjutnya hasil yang didapatkan dari algoritma sebelumnya diolah lagi menggunakan algoritma knapsack problem untuk pembagian ruang ujian. Semua algoritma tersebut diimplementasikan dengan menggunakan bahasa pemrograman *Hypertext Preprocessor* (PHP) yang berfungsi untuk menjadwalkan Ujian Tengah Semester (UTS) dan Ujian Akhir Semester (UAS) di Departemen Teknik Industri dan Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya. *Input* yang digunakan untuk aplikasi berbasis web penjadwalan ujian ini adalah data peserta, data mata kuliah dan data ruang kelas yang diujikan. Sedangkan *output* yang dihasilkan adalah jadwal (waktu dan ruang) ujian.

3.1.6. Uji Coba dan Analisis Performa Algoritma

Pada tahap uji coba algoritma dan analisis performa algoritma, dilakukan evaluasi terhadap hasil pemodelan yang dihasilkan oleh aplikasi berbasis web penjadwalan ujian. Apabila hasilnya sudah memenuhi semua *hard constraints* dan dapat meminimalkan jumlah pelanggaran terhadap *soft constraints*, maka dapat beralih ke proses selanjutnya. Tetapi jika hasilnya belum optimal, maka dapat dilakukan peninjauan kembali terhadap model yang didapatkan. Selain itu pada tahap ini juga dilakukan evaluasi terhadap hasil penjadwalan ujian yang didapatkan. Tujuan yang ingin dicapai adalah membuat jadwal

ujian yang optimal dengan menyediakan waktu istirahat yang cukup bagi mahasiswa, artinya mahasiswa dapat memperoleh jadwal ujian yang tersebar / terdistribusi dengan baik. Disini digunakan nilai proximity untuk mengukur kualitas jadwal yang dihasilkan. Semakin rendah nilai proximitinya, maka dapat dikatakan bahwa semakin baik pula performa dari algoritma yang digunakan.

3.1.7. Analisis Jadwal Ujian

Pada tahap analisis jadwal ujian, penulis melakukan perbandingan antara hasil penjadwalan dengan algoritma *great deluge* dibandingkan dengan penjadwalan secara manual dan penjadwalan dengan metode dari penelitian sebelumnya : *simulated annealing*. Serta apakah jadwal yang dihasilkan sudah memenuhi semua *hard constraints* dan *soft constraints* yang ada.

3.1.8. Penyusunan Buku Tugas Akhir

Tahap ini merupakan tahap pendokumentasian hasil tugas akhir serta analisis terhadap hasil akhir dari penelitian. Keluaran dari tahap ini adalah buku tugas akhir. Buku tugas akhir ini diharapkan dapat menjadi referensi bagi penelitian selanjutnya.

(Halaman ini sengaja dikosongkan)

BAB IV PERANCANGAN

Dalam bab mengenai tahap perancangan dari penelitian tugas akhir. Bab ini berisikan hasil pengumpulan data, praproses data, serta pengolahan dan pemodelan data.

4.1. Hasil Pengumpulan Data

Data yang digunakan dalam pengerjaan tugas akhir ini adalah data peserta, data mata kuliah dan data ruang ujian pada semester ganjil tahun ajaran 2017/2018 di Departemen Teknik Industri dan semester genap tahun ajaran 2016/2017 di Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya program studi Sarjana (S1).

Tabel 4.1 Ringkasan data

Jumlah Mata Kuliah	41
Jumlah Mahasiswa	687
Jumlah Kelas	12
Jumlah Sesi	15

4.1.1. Data Peserta Ujian

Data peserta ujian mahasiswa Departemen Teknik Industri pada semester ganjil tahun ajaran 2017/2018 sebesar 687 mahasiswa. Data yang diperoleh berupa NRP mahasiswa, mata kuliah yang diambil beserta jumlah SKS dan kelasnya.

Tabel 4.2 Potongan data peserta ujian

NRP	Mata Kuliah	Sks	Kelas
2510100082	Analisa Keputusan	3	A
	Kerja Praktek	2	A
	Perancangan Fasilitas	3	B
	Tugas Akhir	6	A

NRP	Mata Kuliah	Sks	Kelas
2511100027	Computer Integrated Manufacturing	3	A
	Faal dan Biomenikanika Kerja	3	A
	Pemeliharaan dan Teknik Keandalan	3	B
	Penelitian Operasional II	3	B
	Perancangan Fasilitas	3	B
	Perencanaan Industri II	3	B
	Technopreneurship	3	12
	Teknik Pengendalian Kualitas	3	B
2511100172	Kerja Praktek	2	A
	Perencanaan Industri II	3	D
	Tugas Akhir	6	A

4.1.2. Data Mata Kuliah

Data mata kuliah berisikan semua mata kuliah yang dibuka pada semester ganjil tahun ajaran 2017/2018 serta jumlah peserta didik yang mengambil mata kuliah tersebut.

Tabel 4.3 Data mata kuliah

Mata Kuliah	Jumlah Mahasiswa
Analisa Keputusan	24
Analisa Produktivitas	20
Analisis dan Estimasi Biaya	178
Aplikasi Ergonomi Industri	22
Computer Integrated Manufacturing	10
Data Mining	22
Ergo Safety	24
Ergonomi Industri	183
Faal dan Biomenikanika Kerja	7

Mata Kuliah	Jumlah Mahasiswa
Makro Ergonomi	20
Manajemen Distribusi	24
Manajemen Lingkungan Industri	22
Manajemen Pengetahuan	23
Manajemen Persediaan dan Pengadaan	22
Manajemen Resiko Korporat	23
Matematika Optimasi	207
Mekanika Teknik	25
Menggambar Teknik	194
Metodologi Penyelesaian Masalah	23
Metodologi Sistem Dinamik	12
Otomasi Industri	189
Pemeliharaan dan Teknik Keandalan	152
Penelitian Operasional II	144
Pengantar Ilmu Ekonomi	200
Pengantar Teknik dan Sistem Industri	202
Pengetahuan Bahan Teknik	194
Perancangan dan Pengembangan Produk	134
Perancangan Fasilitas	142
Perancangan Sistem Informasi Bisnis	160
Perencanaan dan Pengendalian Produksi	170
Perencanaan Industri II	147
Permodelan Sistem Berbasis Agen	18
Proses Manufaktur	107
Rekayasa Proses Bisnis	22
Simulasi Diskrit Terapan	14
Six Sigma	22
Statistik Industri II	199

Mata Kuliah	Jumlah Mahasiswa
Supply Chain Management	25
Teknik Pengendalian Kualitas	141
Teori Permainan	16
Termodinamika	94

4.1.3. Data Ruang Ujian

Data ruang ujian meliputi nama ruang yang digunakan untuk ujian beserta kapasitasnya.

Tabel 4.4 Data ruang ujian

No	Kode Ruang	Kapasitas
1	IE - 101	40 - 45 mhs
2	IE - 102	40 - 45 mhs
3	IE - 103	40 - 45 mhs
4	IE - 104	40 - 45 mhs
5	IE - 105	40 - 45 mhs
6	IE - 106	40 mhs
7	IE - 108	40 mhs
8	IE - 109	70 mhs
9	Auditorium Sinarmas	200 mhs
10	Lab. MM	40 - 45 mhs
11	IE - 303 D	15
12	IE - 302	120

4.2. Pra-Proses Data

Pada tahap pra-proses data, dilakukan pengolahan data hasil wawancara ke dalam bentuk data dengan format *Carter dataset*,

sebagai data masukan untuk aplikasi penjadwalan ujian otomatis. Hasil pra-proses data berupa *file* .stu, .crs dan .room.

4.2.1. Data CRS

Data .crs berisikan data kode mata kuliah dan jumlah peserta ujian pada setiap mata kuliah. Data ini didapatkan dari data mata kuliah yang telah dijelaskan pada sub-bab 4.1.2. dimana dari data tersebut dilakukan pengkodean mata kuliah dengan format urutan numerik 4 digit.

Tabel 4.5 Konversi kode mata kuliah

Mata Kuliah	Kode MK
Analisa Keputusan	0001
Analisa Produktivitas	0002
Analisis dan Estimasi Biaya	0003
Aplikasi Ergonomi Industri	0004
Computer Integrated Manufacturing	0005
Data Mining	0006
Ergo Safety	0007
Ergonomi Industri	0008
Faal dan Biomenikanika Kerja	0009
Makro Ergonomi	0010
Manajemen Distribusi	0011
Manajemen Lingkungan Industri	0012
Manajemen Pengetahuan	0013
Manajemen Persediaan dan Pengadaan	0014
Manajemen Resiko Korporat	0015
Matematika Optimasi	0016
Mekanika Teknik	0017
Menggambar Teknik	0018
Metodologi Penyelesaian Masalah	0019
Metodologi Sistem Dinamik	0020

Mata Kuliah	Kode MK
Otomasi Industri	0021
Pemeliharaan dan Teknik Keandalan	0022
Penelitian Operasional II	0023
Pengantar Ilmu Ekonomi	0024
Pengantar Teknik dan Sistem Industri	0025
Pengetahuan Bahan Teknik	0026
Perancangan dan Pengembangan Produk	0027
Perancangan Fasilitas	0028
Perancangan Sistem Informasi Bisnis	0029
Perencanaan dan Pengendalian Produksi	0030
Perencanaan Industri II	0031
Permodelan Sistem Berbasis Agen	0032
Proses Manufaktur	0033
Rekayasa Proses Bisnis	0034
Simulasi Diskrit Terapan	0035
Six Sigma	0036
Statistik Industri II	0037
Supply Chain Management	0038
Teknik Pengendalian Kualitas	0039
Teori Permainan	0040
Termodinamika	0041

Data hasil pengkodean mata kuliah dan jumlah peserta ujian setiap mata kuliah disimpan dalam file dengan format .crs. data kode mata kuliah dan jumlah peserta ujian dipisahkan dengan menggunakan spasi. Potongan dari *file* dengan format .crs dapat dilihat pada gambar 4.1.

1	0001	24
2	0002	20
3	0003	178
4	0004	22
5	0005	10
6	0006	22
7	0007	24
8	0008	183
9	0009	7
10	0010	20
11	0011	24
12	0012	22
13	0013	23
14	0014	22
15	0015	23
16	0016	207

Gambar 4.1 Potongan file .crs

4.2.2. Data STU

Data STU merupakan data yang berisikan mata kuliah yang diambil oleh setiap mahasiswa. Data ini didapatkan dari data mahasiswa yang telah dijelaskan pada sub-bab 4.1.1. dengan menggunakan bantuan *Pivot Table* pada *Microsoft Excel*. Sehingga menghasilkan tabel seperti pada tabel 4.6.

Tabel 4.6 Potongan Hasil Pivot Tabel Data Peserta Ujian

NRP	0001	0002	0003	0004	0005	0006
2514100007			1			1
2514100008	1	1				1
2514100009						1
2514100010		1				
2514100011	1					
2514100012						
2514100013		1				
2514100015		1				

Pada tabel 4.6. menunjukkan hasil *Pivot Table* data peserta ujian, dimana kolom pertama berisikan NRP peserta ujian, kolom kedua dan seterusnya pada *header* berisikan kode mata kuliah. Perpotongan NRP peserta ujian dengan mata kuliah menunjukkan bahwa peserta ujian tersebut mengambil mata kuliah tersebut atau tidak. Angka 1 menunjukkan bahwa peserta ujian dengan NRP *Y* mengambil mata kuliah *X*. Didalam file .stu angka 1 pada *Pivot Table* diganti dengan kode mata kuliah, sedangkan NRP mahasiswa diwakili dengan nomor baris. Potongan file .stu dapat dilihat pada gambar 4.2.

1	0001	0028					
2	0005	0009	0022	0023	0028	0031	0039
3	0031						
4	0032						
5	0009						
6	0039						
7	0004	0019	0022	0029	0031	0037	0039
8	0009	0023	0032	0039			
9	0009	0019					
10	0008	0016	0028	0030	0037		
11	0014	0034					
12	0031						
13	0007	0014					
14	0016	0027	0028	0030	0039		
15	0031						
16	0004	0007	0010	0019	0036		
17	0013	0030					
18	0034						
19	0016	0019	0023	0030	0031	0039	
20	0011						
21	0013	0014	0031				
22	0010	0022	0029	0030	0031	0037	

Gambar 4.2 Potongan file .stu

4.2.3. Data ROOM

Data pada *file* .room berisikan kapasitas pada setiap ruang ujian dan nomor baris pada *file* .room menunjukkan ruang yang digunakan untuk ujian. Data tersebut didapatkan dari data ruang

yang telah dijelaskan pada sub-bab 4.1.3.. Potongan dari file .room dapat dilihat pada gambar 4.3.

1	45
2	45
3	45
4	45
5	45
6	40
7	40
8	70
9	200
10	45
11	15
12	120

Gambar 4.3 Potongan file .room

4.3. Formulasi Fungsi Tujuan

Pada penjadwalan ujian yang dibahas pada tugas akhir ini terdapat *hard constraint* dan *soft constrain* yang perlu dipenuhi. Pada bagian ini dilakukan pemodelan matematika untuk seluruh *constrain* yang dimiliki.

4.3.1. Variabel Keputusan

Variabel keputusan dari permasalahan penjadwalan ujian digunakan untuk memenuhi batasan yang ditentukan sebelumnya, yaitu memastikan bahwa setiap mata kuliah terjadwal (4) mata kuliah i menunjukkan mata kuliah paralel, demo final project dan ujian praktikum. Mata kuliah paralel merupakan mata kuliah i (kelas A), i (kelas B), dst. dijadikan satu yaitu mata kuliah i .

$$X_{itr} = \begin{cases} 1, & \text{jika mata kuliah ke } i \text{ dijadwalkan pada timeslot } t \text{ di ruang } r \\ 0, & \text{jika tidak} \end{cases}$$

(4)

Dimana,

X_{it} = Mata kuliah i yang dijadwalkan pada timeslot t

$i = \{1, 2, 3, \dots, n\}$ adalah urutan mata kuliah

$t = \{1, 2, 3, \dots, k\}$ adalah urutan slot ujian yang tersedia

$r = \{1, 2, 3, \dots, k\}$ adalah urutan ruang ujian

4.3.2. Batasan

Terdapat beberapa batasan (*hard constraints*) yang perlu diperhatikan agar dapat menghasilkan solusi yang *feasible*. Yang pertama yaitu setiap mata kuliah harus dijadwalkan sesuai dengan slot yang tersedia, dinyatakan pada persamaan (5) dimana X_{it} merupakan mata kuliah ke i yang dijadwalkan pada slot ke t .

$$\sum_{i=1}^n \sum_{t=1}^k \sum_{r=1}^m X_{itr} = 1 \quad (5)$$

Selanjutnya yaitu jadwal yang dibuat tidak ada mahasiswa yang mengambil dua mata kuliah yang saling bentrok satu sama lain, yang dimodelkan dengan persamaan (6)

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij} V_{ij} = 0$$

$$V_{ij} = \begin{cases} 1, & \text{jika } t_i = t_j \\ 0, & \text{lainnya} \end{cases}$$

(6)

t_i = slot ujian mata kuliah i

C_{ij} = jumlah mahasiswa yang mengikuti matakuliah i dan j

V_{ij} = *vector* yang menunjukkan apakah mata kuliah i dan mata kuliah j dijadwalkan pada timeslot (hari dan jam) yang sama.

n = Jumlah mata kuliah.

Batasan yang ketiga yaitu jumlah mahasiswa peserta ujian tidak melebihi kapasitas ruang ujian dalam satu sesi yang ditunjukkan

pada persamaan (7) dimana S_i merupakan jumlah mahasiswa peserta ujian untuk mata kuliah ke- i .

$$\sum_{i=1}^n S_i X_{it} \leq \text{kapasitas total ruangan}$$

(7)

Batasan yang keempat merupakan batasan bahwa jumlah mahasiswa peserta ujian tidak melebihi kapasitas setiap ruang ujian yang ditunjukkan pada persamaan (8) Dimana R_i merupakan kapasitas ruangan i pada timeslot t .

$$\sum_{i=1}^n S_i X_{itr} \leq R_{it}$$

(8)

4.3.3. Fungsi Tujuan

Fungsi tujuan dari permasalahan penjadwalan ujian untuk memastikan bahwa jadwal yang dihasilkan optimal atau tidak yaitu dengan meminimalkan proximity cost, P , yang dinyatakan dalam persamaan matematis (9).

$$P = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij} W_{|t_j-t_i|}}{M}$$

$$W_{|t_j-t_i|} = \begin{cases} 2^{5-|t_j-t_i|}, & \text{jika } 1 \leq |t_j-t_i| \leq 5 \\ 0, & \text{jika } |t_j-t_i| > 5 \end{cases}$$

(9)

Dimana,

C_{ij} = jumlah mahasiswa yang mengikut matakuliah i dan j
 $W_{|t_j-t_i|}$ = bobot selisih slot ujian matakuliah i dan j

M = jumlah seluruh mahasiswa

4.3.4. Matriks Konflik (C_{ij})

Matriks C_{ij} menunjukkan matriks dua dimensi dengan jumlah baris dan jumlah kolom sebanyak jumlah mata kuliah yaitu 41 mata kuliah. Matriks ini berfungsi untuk mengetahui matakuliah mana saja yang berpotensi bertabrakan saat dijadwalkan. Perpotongan antar baris dan kolom menunjukkan jumlah mahasiswa yang mengambil mata kuliah i dan mata kuliah j . potongan matriks konflik dapat dilihat pada tabel 4.7.

Tabel 4.7 Matriks Konflik

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012
0001	0	3	0	1	1	5	1	0	0	0	5	2
0002	3	0	0	3	0	4	4	0	0	3	0	5
0003	0	0	0	3	0	1	0	156	0	1	0	1
0004	1	3	3	0	0	0	6	0	1	7	0	4
0005	1	0	0	0	0	2	1	0	1	0	1	0
0006	5	4	1	0	2	0	0	0	0	0	6	2
0007	1	4	0	6	1	0	0	0	0	5	2	6
0008	0	0	156	0	0	0	0	0	0	0	0	0
0009	0	0	0	1	1	0	0	0	0	1	0	0
0010	0	3	1	7	0	0	5	0	1	0	0	1
0011	5	0	0	0	1	6	2	0	0	0	0	1
0012	2	5	1	4	0	2	6	0	0	1	1	0

4.3.5. Matriks Bobot Selisih Slot (W)

Matriks bobot selisih slot merupakan matriks dua dimensi dengan jumlah baris dan jumlah kolom sesuai dengan jumlah mata kuliah yang di ujikan. Matriks ini digunakan untuk mencari nilai *proximity cost* . Potongan matriks bobot selisih slot dapat dilihat pada tabel 4.8.

Tabel 4.8 Matriks Selisih Bobot Slot

0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012
0001	0	16	0	1	0	0	0	8	8	0	8	4
0002	16	0	0	0	0	0	0	16	4	0	16	2
0003	0	0	0	16	0	4	4	0	2	0	0	4
0004	1	0	16	0	16	2	2	0	4	16	0	8
0005	0	0	0	16	0	4	4	0	2	0	0	4
0006	0	0	4	2	4	0	0	0	0	4	0	0
0007	0	0	4	2	4	0	0	0	0	4	0	0
0008	8	16	0	0	0	0	0	0	2	0	0	1
0009	8	4	2	4	2	0	0	2	0	2	2	16
0010	0	0	0	16	0	4	4	0	2	0	0	4
0011	8	16	0	0	0	0	0	0	2	0	0	1
0012	4	2	4	8	4	0	0	1	16	4	1	0

4.4. Implementasi Algoritma *Graph Colouring*

Algoritma *graph colouring* digunakan untuk menghasilkan jadwal baru sebagai *initial solution* berdasarkan data praproses yang telah dimasukkan. Jadwal baru ini selanjutnya dioptimasi dengan menggunakan algoritma *great deluge*.

4.5. Implementasi Algoritma *Great Deluge*

Implementasi algoritma *great deluge* digunakan untuk mengoptimasi *initial solution* yang telah dihasilkan oleh algoritma sebelumnya. Pada algoritma *great deluge* menggunakan dua metode untuk pengoptimasian jadwal, yaitu dengan cara mengganti slot mata kuliah secara acak dan menukar slot dua mata kuliah yang dipilih secara acak.

(Halaman ini sengaja dikosongkan)

BAB V IMPLEMENTASI

Pada bab ini membahas mengenai proses pembuatan *solver* penjadwalan ujian otomatis dengan menggunakan Bahasa pemrograman *Hypertext Preprocessor* (PHP).

5.1. Membaca Input File

Terdapat tiga macam *input file* yang digunakan sebagai masukan data awal yaitu *file* dengan format .crs , .stu, dan .room. Pada potongan kode 5.1 menunjukkan bahwa *file* crs disimpan ke dalam variabel \$txt_filecrs setelah itu dilakukan pemisahan pada setiap baris dan dimasukkan ke dalam *array* \$rowscrs.

```
//mengambil file
$txt_filecrs    = file_get_contents($target_file0);
$txt_filestu    = file_get_contents($target_file1);
$txt_fileroom   = file_get_contents($target_file2);

//Memisah antar enter
$rowscrs        = explode("\n", $txt_filecrs);
$rowsstu        = explode("\n", $txt_filestu);
$rowsroom       = explode("\n", $txt_fileroom);

$sumcrs         = count($rowscrs);
$sumstu         = count($rowsstu);
$sumroom        = count($rowsroom);
$sumtmlst= $_POST['Timeslot'];
```

Kode 5.1 Membaca file

5.1.1. File CRS

File dengan format .crs berisi data mata kuliah beserta jumlah peserta ujian pada masing-masing mata kuliah. Pada potongan kode 5.2 menunjukkan bahwa jumlah mahasiswa peserta ujian

dimasukkan ke dalam *array* \$kapasitas, sedangkan \$i menunjukkan urutan mata kuliah.

```
//Membaca file .crs
for($i =0;$i < $sumcrs;$i++){
    $datakapasitas = explode(' ', $rowscrs[$i]);
    $kapasitas[$i] = $datakapasitas[1];
}
```

Kode 5.2 Membaca file crs

5.1.2. File STU

File dengan format .stu berisi data mata kuliah yang diambil oleh setiap masing-masing mahasiswa. Pada potongan kode 5.3 dilakukan pemisahan antar mata kuliah didalam setiap baris, selanjutnya mata kuliah tersebut dimasukkan ke dalam *array* dua dimensi, dimana dimensi pertama menunjukkan mahasiswa peserta ujian dan dimensi kedua menunjukkan mata kuliah yang diambil oleh mahasiswa peserta ujian tersebut.

```
//membaca file .stu dan memasukkannya ke array
for($i =0;$i < $sumstu;$i++){
    $data = explode(' ', $rowssstu[$i]);
    for ( $j = 0; $j < count($data); $j++ ) {
        if($data[$j] > 0){
            $hasil[$i][$j] = $data[$j];
        }
    }
}
```

Kode 5.3. Membaca file stu

5.1.3. File ROOM

File dengan format .room berisi kapasitas masing-masing tiap ruang. Didalam potongan kode 5.4 menunjukkan bahwa kapasitas tiap ruang di masukkan ke dalam *array* \$kapruang. Setelah itu dilakukan perhitungan kapasitas ruang total seperti yang ada pada potongan kode 5.5.

```
//Membaca file dari .room
for($i =0;$i < $sumroom;$i++){
    $kapruang[$i] = $rowsroom[$i];
}
```

Kode 5.4. Membaca file room

```
//Menghitung kapasitas Ruang Total
$KapasitasTotal =0;
for($i =0;$i < $sumroom;$i++){
    $KapasitasTotal = $KapasitasTotal + intval($rowsroom[$i]);
}
```

Kode 5.5. Menghitung kapasitas total ruang

5.2. Pembuatan Matriks Konflik

Konflik matriks digunakan untuk mengetahui jumlah mahasiswa yang mengikuti dua mata kuliah. Matriks konflik terdiri dari dua dimensi dengan panjang dimensi masing-masing sesuai dengan jumlah mata kuliah hasil dari pembacaan file .stu.

```
//mencari conflict matriks
for ($i = 0; $i < $sumstu; $i++) {
    $data = explode(' ', $rowsstu[$i]);
    for ($j = 0; $j < count($data)-1; $j++) {
        for ($k = $j+1; $k < count($data); $k++) {
            $stij = $hasil[$i][$j];
            $stik = $hasil[$i][$k];
            $conflict[intval($stij)-1][intval($stik)-1]++;
            $conflict[intval($stik)-1][intval($stij)-1]++;
        }
    }
}
```

Kode 5.6. Membuat konflik matriks

5.3. Penerapan Algoritma *Graph Colouring*

Algoritma *graph colouring* digunakan untuk mencari *initial solution* yang nantinya akan menjadi masukan untuk optimasi yang dilakukan algoritma *great deluge*. Didalam pengerjaan tugas akhir ini penulis menggunakan teknik heuristik *Largest Degree* yaitu mengurutkan mata kuliah berdasarkan jumlah

konflik terbesar. Untuk menentukan mata kuliah yang konflik dengan mata kuliah lain dapat dilihat pada potongan kode 5.7. Array \$jadwal menunjukkan array dua dimensi dengan jumlah baris sebanyak jumlah mata kuliah, sedangkan jumlah kolom sesuai dengan banyak konflik dengan mata kuliah lain. Setelah itu dilakukan pengurutan jumlah konflik yang terbesar dengan fungsi *rsort* pada Bahasa pemrograman PHP.

```
//Jadwal Konflik
for ($i = 0; $i < $sumcrs; $i++) {
    $temp=array();
    for($j = 0; $j <$sumcrs; $j++){
        $jadwalBisa = $conflict[$i][$j];
        if($jadwalBisa >=1 && $i != $j){
            array_push($temp,$j);
        }
    }
    $jadwal[$i][0] =$i;
    for ($l = 0; $l < count($temp) ; $l++) {
        $jadwal[$i][$l+1] = $temp[$l];
    }
}
rsort($jadwal); //untuk mengurutkan jadwal dengan degree yang paling besar
```

Kode 5.7. Jadwal konflik

Untuk menentukan suatu mata kuliah dijadwalkan pada slot waktu tertentu digunakan dua macam *function* yaitu *function* cekSlot dan cekRoom. *Function* cekSlot digunakan untuk membandingkan suatu mata kuliah tidak bentrok dengan mata kuliah yang konflik dengannya dalam satu slot.

```
//Method cek Slot
function cekSlot($indeks,$slot,$jadwal){
    global $timeslot;
    for ($i = 1; $i < count($jadwal[$indeks]); $i++) {
        $matkul = $jadwal[$indeks][$i];
        if($timeslot[$matkul]==$slot){
            return false;
        }
    }
    return true;
}
```

Kode 5.8. Method cekSlot

Function cekRoom digunakan untuk memastikan bahwa jumlah mahasiswa peserta ujian dalam satu slot waktu tidak melebihi jumlah kapasitas seluruh ruangan. Pada potongan kode 5.9 menunjukkan apabila jumlah mahasiswa peserta ujian suatu mata kuliah lebih sedikit dari jumlah kapasitas pada suatu slot maka mata kuliah tersebut dimasukkan ke dalam slot tersebut dan kapasitas pada slot tersebut akan dikurangi sejumlah mahasiswa peserta ujian mata kuliah tersebut.

```
//Method cek Room
function cekRoom($matkul,$slot,$kapasitas) {
    global $kapasitastiptimeslot;
    $a = $kapasitas[$matkul];
    if($a <= $kapasitastiptimeslot[$slot-1]){
        $kapasitastiptimeslot[$slot-1] = $kapasitastiptimeslot[$slot-1] - $a;
        return true;
    }
    return false;
}
```

Kode 5.9. Method cekRoom

Function explore pada potongan kode AAA digunakan untuk membuat jadwal awal dengan memasukkan nomor *slot* pada setiap mata kuliah. Apabila suatu mata kuliah telah memenuhi kedua *function* *cekSlot* dan *cekRoom* maka suatu mata kuliah dapat dijadwalkan pada slot tersebut. Sedangkan jika mata kuliah tidak memenuhi kedua *function* diatas maka dilakukan pengecekan kembali dengan menggunakan slot berikutnya.

```
//Method Explore
function explore($matkul, $slot, $jadwal, $kapasitas){
    global $timeslot;
    global $sumcrs;
    global $sumtmslt;
    global $kapasitastiptimeslot;
    if($matkul >= count($jadwal)){
        return true;
    }
    $MK = $jadwal[$matkul][0];
    if (cekSlot($matkul,$slot,$jadwal) && cekRoom($MK,$slot,$kapasitas)){
        $timeslot[$MK] = $slot;
        for ($i = 1; $i <= $sumtmslt; $i++){
            if (explore($matkul+1, $i, $jadwal, $kapasitas)){
                return true;
            }
        }
    }
    return false;
}
```

Kode 5.10. Method explore

5.4. Penerapan Algoritma *Great Deluge*

Penerapan algoritma *great deluge* bertujuan untuk melakukan optimasi *initial solution* yang telah didapat dari algoritma *graph colouring*. Beberapa variabel yang perlu diinisialisasi pada algoritma *great deluge* yaitu nilai *proximity* dari *initial solution* (fs), nilai *proximity* yang diinginkan (D), jumlah iterasi (I), level (B) dan pengurangan level (deltaB). Untuk level diinisiasikan sama dengan nilai *proximity initial solution*.

```
function GreatDeluge($Proximity, $S){
    global $sumcrs;
    $fs=$Proximity;
    $Sbest = $S;
    $fsbest=$fs;
    $D = 30;
    $I =1000000;
    $B = $fs;
    $deltaB = ($fs-$D)/$I;
    $iterasi = 0;
    $not_improving_length_GDA = 100;
    $not_improving_counter= 0;
```

Kode 5.11. Method Great Deluge

Jadwal yang telah dihasilkan dari algoritma *graph colouring* akan diubah untuk mendapat jadwal yang paling optimal. Untuk mengubah jadwal tersebut penulis menggunakan dua

pendekatan *low level heuristic* yaitu *random move* dan *random swap*.

```
$angkarandom = rand();
if ($angkarandom % 2 ==1) {
    $Sbaru=move($S);
} else {
    $Sbaru=swap($S);
}
```

Kode 5.12. Pemilihan low level heuristic

Function move yaitu mengubah jadwal sebelumnya dengan memindahkan salah satu mata kuliah acak ke slot waktu secara acak. Selanjutnya dilakukan pengecekan pada *function* GDcekSlot dan cekRoom untuk memastikan bahwa jadwal yang baru tidak bentrok dan tidak melebihi kapasitas ruang.

```
function move($timeslot){
    global $sumcrs, $sumtmslt;
    global $kapasitas;
    $jadwalbaru = $timeslot;
    $randomMK = rand(0,($sumcrs-1));
    $randomTS = rand(1,$sumtmslt);

    while($timeslot[$randomMK]==$randomTS){
        $randomTS = rand(1,$sumtmslt);
    }

    if (GDcekSlot($randomMK,$randomTS,$jadwalbaru) &&
        cekRoom($randomMK,$randomTS,$kapasitas)){
        $jadwalbaru[$randomMK]=$randomTS;
    }
    return $jadwalbaru;
}
```

Kode 5.13. Method move

Function swap digunakan untuk mengubah jadwal dengan mengambil mata kuliah secara acak lalu ditukar timeslotnya. Didalam *function* swap, jadwal yang telah ditukar dibandingkan dengan menggunakan *function* GDcekSlot dan cekRoom untuk memastikan bahwa jadwal yang baru memenuhi dua *function* tersebut.

```

function swap($timeslot){
    global $sumcrs;
    global $kapasitas;
    $jadwalbaru = $timeslot;
    $angkarandom = rand(0, ($sumcrs-1));
    $angkarandom2 = rand(0, ($sumcrs-1));

    while ($angkarandom==$angkarandom2) {
        $angkarandom2=rand(0, ($sumcrs-1));
    }

    if (GDcekSlot($angkarandom,$jadwalbaru[$angkarandom2],$jadwalbaru) &&
        GDcekSlot($angkarandom2,$jadwalbaru[$angkarandom],$jadwalbaru) &&
        cekRoom($angkarandom,$jadwalbaru[$angkarandom2],$kapasitas) &&
        cekRoom($angkarandom2,$jadwalbaru[$angkarandom],$kapasitas)
    ){
        $temp = $jadwalbaru[$angkarandom];
        $jadwalbaru[$angkarandom] = $jadwalbaru[$angkarandom2];
        $jadwalbaru[$angkarandom2] = $temp;
    }
    return $jadwalbaru;
}

```

Kode 5.14. Method swap

Jadwal yang telah diubah dengan menggunakan dua pendekatan *low level heuristic* move dan swap selanjutnya akan dibandingkan dengan nilai *proximity* jadwal terbaik sebelumnya, apabila nilai jadwal yang baru kurang dari nilai *proximity* jadwal terbaik sebelumnya atau kurang dari level, maka jadwal tersebut diterima.


```

while($iterasi<$I){
    $angkarandom = rand();
    if ($angkarandom % 2 ==1){
        $Sbaru=move($S);
    } else {
        $Sbaru=swap($S);
    }
    $fsbaru = hitungProximity($Sbaru);
    if ($fsbaru<$fsbest){
        $S=$Sbaru;
        $fs=$fsbaru;
        $Sbest=$Sbaru;
        $fsbest=$fsbaru;
        $not_improving_counter= 0;
    } else if ($fsbaru<=$B){
        $S = $Sbaru;
        $fs = $fsbaru;
        $not_improving_counter= 0;
    } else {
        $not_improving_counter++;
        if($not_improving_counter == $not_improving_length_GDA){
            $S = $Sbest;
            $fs = $fsbest;
            $B = $fs;
            $deltaB = ($fs-$D)/($I-$iterasi);
            $not_improving_counter= 0;
        }
    }
    $B=$B-$deltaB;
    $iterasi++;
}

```

Kode 5.15. Great Deluge

5.5. Penerapan Algoritma Knapsack Problem

Algoritma *Knapsack Problem* digunakan untuk membagi lokasi ujian per ruangan dalam setiap slot. Pada penelitian ini suatu ruangan ujian dapat ditempati oleh peserta ujian dengan beberapa mata kuliah yang berbeda, sehingga pada potongan kode 5.16 menunjukkan bahwa apabila suatu ruang ujian masih terdapat sisa tempat, maka akan diisi dengan mata kuliah selanjutnya.

```

function bagiKelas($TS){
global $sumtmslt, $kapruang, $kapasitas;
for ($i = 1; $i <= $sumtmslt; $i++) {
    $studenttimeslot=0; $k=0; $temp = $kapruang; $m=0;
    for($j = 0; $j < count($TS[$i-1]); $j++){
        $studenttimeslot = $kapasitas[$TS[$i-1][$j]];
        while ($studenttimeslot>0){
            if($studenttimeslot > (int)$temp[$k]){
                $jmlhmshtiapruang[$TS[$i-1][$j]][$i][$k] = $temp[$k];
                $matkultsruang[$i][$k][$m]=$TS[$i-1][$j];
                $studenttimeslot -= $temp[$k];
                $temp[$k] -= $temp[$k];
                $k++;
            }
            else {
                $jmlhmshtiapruang[$TS[$i-1][$j]][$i][$k] = $studenttimeslot;
                $matkultsruang[$i][$k][$m]=$TS[$i-1][$j];
                $temp[$k] -= $studenttimeslot;
                $studenttimeslot -= $studenttimeslot;
                $m++;
            }
        }
    }
}
return $matkultsruang; }

```

Kode 5.16. Method bagiKelas

5.6. Tampilan Aplikasi

Berikut ini merupakan tampilan aplikasi yang dihasilkan dari proses implementasi kode pada sub-bab sebelumnya.

5.6.1. Halaman Utama

Pada halaman utama aplikasi, pengguna di haruskan melakukan *upload* beberapa *file* dan mengisikan jumlah slot waktu yang diinginkan. *File* yang harus di *upload* berupa *file* dengan format .crs, .stu dan .room. tampilan halaman utama aplikasi dapat dilihat pada gambar berikut.

**SISTEM INFORMASI
PENJADWALAN UJIAN**

Upload file

File CRS :
 Tidak ada file yang dipilih

File STU :
 Tidak ada file yang dipilih

File ROOM :
 Tidak ada file yang dipilih

Jumlah Timeslot :

Gambar 5.1 Tampilan Halaman Utama Aplikasi

5.6.2. Halaman Hasil Penjadwalan

Halaman ini menampilkan hasil dari penjadwalan ujian yang telah mendapat *input* data dari halaman utama. Terdapat 4 kotak hijau yang menampilkan jumlah mata kuliah yang di ujikan, jumlah mahasiswa peserta ujian, jumlah slot waktu yang diinginkan dan jumlah ruang ujian. Selain itu pada halaman hasil penjadwalan juga terdapat nilai *proximity cost* yang menunjukkan seberapa optimal jadwal yang telah di optimasi oleh algoritma *great deluge*.

SISTEM INFORMASI PENJADWALAN UJIAN												
41 Jumlah Mata Kuliah			687 Jumlah Mahasiswa			15 Jumlah Timeslot			12 Jumlah Ruang			
Hari	Jam	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Suarman	Lab. MM	IE - 303 D
Senin	08.00- 10.00	Otomasi Industri (1 - 45)	Otomasi Industri (46 - 90)	Otomasi Industri (91 - 135)	Otomasi Industri (136 - 180)	Otomasi Industri (181 - 189) Perencanaan Industri II (1 - 36)	Perencanaan Industri II (37 - 76)	Perencanaan Industri II (77 - 116)	Perencanaan Industri II (117 - 147)			

Gambar 5.2 Halaman Hasil Penjadwalan (1)

Halaman hasil penjadwalan juga menampilkan jadwal ujian dengan tanpa ruang (gambar 5.2) dan jadwal ujian dengan menggunakan ruang (gambar 5.3).

Hari	Jam	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109
Senin	08.00- 10.00	Otomasi Industri (1 - 45)	Otomasi Industri (46 - 90)	Otomasi Industri (91 - 135)	Otomasi Industri (136 - 180)	Otomasi Industri (181 - 189) Perencanaan Industri II (1 - 36)	Perencanaan Industri II (37 - 76)	Perencanaan Industri II (77 - 116)	Perencanaan Industri II (117 - 147)
Senin	10.30- 12.30	Pengantar Teknik dan Sistem Industri (1 - 45)	Pengantar Teknik dan Sistem Industri (46 - 90)	Pengantar Teknik dan Sistem Industri (91 - 135)	Pengantar Teknik dan Sistem Industri (136 - 180)	Pengantar Teknik dan Sistem Industri (181 - 202) Six Sigma (1 - 22)	Teknik Pengendalian Kualitas (23 - 62)	Teknik Pengendalian Kualitas (63 - 102)	Teknik Pengendalian Kualitas (103 - 163)
Senin	13.30- 15.30	Analisa Keputusan (1 - 24) Manajemen Persediaan dan Pengadaan (1 - 21)	Manajemen Persediaan dan Pengadaan (22 - 22) Termodinamika (1 - 44)	Termodinamika (45 - 89)	Termodinamika (90 - 94)				

Gambar 5.3 Halaman Hasil Penjadwalan (2)

5.6.3. Output File Aplikasi

Hasil luaran yang didapatkan dari hasil optimasi penjadwalan ujian disimpan dalam bentuk txt. Terdapat dua luaran *file* txt yaitu *file* yang berisikan informasi mengenai optimasi penjadwalan dan *file* kedua berisikan jadwal ujian yang sudah jadi.

```

1  Algoritma Great Deluge
2  Estimated Quality : 30
3  Inisiasi Level : fs - 56.599708879185
4  not_improving_length_GDA : 100
5  Execution Time : 944.44053602219 detik
6  Jumlah Iterasi : 1000000
7  Hasil Penjadwalan : 4 2 9 8 12 8 9 15 3 10 15
8  Nilai Proximity : 29.465793304221

```

Gambar 5.4 Output file Aplikasi (1)

Pada *file* kedua, data yang didapat berupa hari ujian, jam ujian mata kuliah ujian dan ruang ujian.

```

1  Senin,08.00-10.00,Pengantar Ilmu Ekonomi
2  ,IE - 101
3  Senin,08.00-10.00,Pengantar Ilmu Ekonomi
4  ,IE - 102
5  Senin,08.00-10.00,Pengantar Ilmu Ekonomi
6  ,IE - 103
7  Senin,08.00-10.00,Pengantar Ilmu Ekonomi
8  ,IE - 104
9  Senin,08.00-10.00,Pengantar Ilmu Ekonomi
10 ;Perancangan Sistem Informasi Bisnis
11 ,IE - 105
12 Senin,08.00-10.00,Perancangan Sistem Informasi Bisnis
13 ,IE - 106
14 Senin,08.00-10.00,Perancangan Sistem Informasi Bisnis
15 ,IE - 108
16 Senin,08.00-10.00,Perancangan Sistem Informasi Bisnis
17 ,IE - 109
18 Senin,10.30-12.30,Statistik Industri II
19 ,IE - 101

```

Gambar 5.5 Output file Aplikasi (2)

(Halaman ini sengaja dikosongkan)

BAB VI

HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil yang didapatkan dari pengerjaan tugas akhir ini serta pembahasan secara keseluruhan yang didapatkan dari penelitian.

6.1. Data Uji Coba

Data yang digunakan dalam pengerjaan tugas akhir ini adalah data mahasiswa peserta ujian, mata kuliah yang diujikan dan ruang ujian Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya pada semester ganjil tahun ajaran 2017/2018. Pada data ini diketahui terdapat 687 mahasiswa, 41 mata kuliah dan 12 ruang ujian.

6.2. Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak. Spesifikasi kriteria perangkat keras yang digunakan ini dapat dilihat pada tabel 6.1.

Tabel 6.1 Spesifikasi perangkat keras

Perangkat Keras	Spesifikasi
Jenis	Laptop
Processor	Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30 GHz
RAM	4.00 GB
Hardisk	1 TB

Sedangkan untuk spesifikasi perangkat lunak yang digunakan dalam penelitian ini, ditunjukkan pada tabel 6.2.

Tabel 6.2 Spesifikasi perangkat lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	Windows 10
Bahasa Pemograman	PHP
Tools	XAMPP, Notepad++, Microsoft Excel, Mozilla Firefox

6.3. Fungsi Tujuan dari Jadwal Manual

Fungsi tujuan yang digunakan dalam pengerjaan tugas akhir ini yaitu meminimalkan nilai *proximity cost*. Nilai *proximity cost* dapat dihitung dengan menggunakan persamaan matematika yang sudah dijelaskan pada sub-bab 4.3.3, sebagai berikut :

$$P = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n C_{ij} W_{|t_j-t_i|}}{M}$$

Dari hasil perhitungan, diketahui bahwa nilai *proximity cost* jadwal yang dibuat secara manual yaitu 51.1499. Namun pada jadwal yang dibuat secara manual masih terdapat jadwal yang bertabrakan, dimana terdapat beberapa mahasiswa yang harus melakukan lebih dari satu ujian dalam waktu bersamaan. Jadwal manual dapat dilihat pada lampiran A.

Jadwal yang Bentrok
Mata Kuliah 2 9 = Not Feasible
Mata Kuliah 2 31 = Not Feasible
Mata Kuliah 3 22 = Not Feasible
Mata Kuliah 6 38 = Not Feasible
Mata Kuliah 12 38 = Not Feasible
Mata Kuliah 18 27 = Not Feasible
Mata Kuliah 20 26 = Not Feasible
Mata Kuliah 28 37 = Not Feasible

Gambar 6.1 Permasalahan pada jadwal manual

Tabel 6.3 Jadwal bentrok

No	Mata Kuliah 1	Mata Kuliah 2	Mahasiswa
1	Analisis dan Estimasi Biaya	Makro Ergonomi	2513100150
2	Analisis dan Estimasi Biaya	Permodelan Sistem Berbasis Agen	2513100128, 2514100093
3	Aplikasi Ergonomi Industri	Penelitian Operasional II	2513100150, 2513100157

No	Mata Kuliah 1	Mata Kuliah 2	Mahasiswa
4	Ergo Safety	Teknik Pengendalian Kualitas	2514100103
5	Manajemen Pengetahuan	Teknik Pengendalian Kualitas	2513100178
6	Metodologi Penyelesaian Masalah	Perancangan Fasilitas	2514100113
7	Otomasi Industri	Perancangan dan Pengembangan Produk	2517108708, 2517108709
8	Perancangan Sistem Informasi Bisnis	Supply Chain Management	14 Mahasiswa

Gambar 6.2 menunjukkan hasil penjadwalan manual yang telah dibuat oleh Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember.

Hari & Tanggal	Jam	Mata Kuliah
Senin 16 Oktober 2017	08.00 - 10.00	Menggambar Teknik
		Menggambar Teknik
		Menggambar Teknik
		Menggambar Teknik
		Menggambar Teknik
		Manajemen Persediaan dan Pengadaan
		Manajemen Distribusi
		Ergonomi Industri
		Ergonomi Industri
		Ergonomi Industri
		Ergonomi Industri
		Ergonomi Industri
	10.30 - 12.30	Analisa Produktivitas
		Teori Permainan
		Pengantar Teknik dan Sistem Industri
		Pengantar Teknik dan Sistem Industri

Gambar 6.2 Hasil Penjadwalan Ujian Manual

6.4. Hasil Uji Coba Initial Solution

Pada penelitian ini dalam menemukan jadwal awal sebagai inisiasi solusi menggunakan algoritma *greedy* dan algoritma

graph colouring. Kedua algoritma tersebut dibandingkan mana yang dapat memenuhi seluruh batasan yang ada.

Algoritma *greedy* membuat inisiasi solusi dengan menjadwalkan mata kuliah berdasarkan urutan dari *file* .crs, sedangkan algoritma *graph colouring* menggunakan teknik heuristik *Largest Degree* dalam menjadwalkan inisiasi solusi, yaitu dengan menjadwalkan mata kuliah berdasarkan jumlah konflik terbesar.

Jadwal yang dihasilkan oleh algoritma *greedy* tidak terjadwal sepenuhnya, terdapat beberapa mata kuliah yang belum terjadwal. Hasil dari penjadwalan algoritma *graph colouring* semua mata kuliah terjadwal semua pada timeslot yang ada. Hasil penjadwalan ujian menggunakan algoritma *graph colouring* dapat dilihat pada lampiran B dan lampiran D. Perbandingan jadwal ujian hasil penjadwalan manual, algoritma *greedy* dan algoritma *graph colouring* dapat dilihat pada tabel 6.4.

Tabel 6.4 Perbandingan Hasil Penjadwalan Manual, *Greedy*, dan *Graph Colouring*

Kode MK	Slot		
	Manual	<i>Greedy</i>	<i>Graph</i>
0001	3	1	10
0002	2	2	13
0003	9	1	10
0004	8	3	9
0005	5	15	2
0006	12	4	15
s0007	12	4	15
0008	1	2	5
0009	5	8	6
0010	9	15	14
0011	1	7	9

Kode MK	Slot		
	Manual	<i>Greedy</i>	<i>Graph</i>
0012	6	5	12
0013	12	3	2
0014	1	7	6
0015	9	15	14
0016	15	4	6
0017	12	14	9
0018	1	1	5
0019	6	6	12
0020	5	8	11
0021	5	5	3
0022	4	9	5
0023	8	10	7
0024	7	6	7
0025	2	7	4
0026	3	15	2
0027	5	8	12
0028	6	11	9
0029	13	12	1
0030	11	13	4
0031	7	14	3
0032	9	15	8
0033	8	15	1
0034	6	6	12
0035	10	13	4
0036	5	11	11
0037	10	tidak terjadwal	2
0038	13	tidak terjadwal	8
0039	12	tidak terjadwal	11
0040	2	tidak terjadwal	7

Kode MK	Slot		
	Manual	<i>Greedy</i>	<i>Graph</i>
0041	14	tidak terjadwal	8

Proximity cost yang dihasilkan oleh algoritma *graph colouring* sebesar 56.5997. Hal ini menunjukkan bahwa hasil penjadwalan inisiasi solusi menggunakan algoritma *graph colouring* lebih baik daripada penjadwalan manual dan penjadwalan menggunakan algoritma *greedy*, karena semua mata kuliah terjadwal dan tidak ada mata kuliah yang saling bertabrakan, meskipun nilai *proximity cost* lebih tinggi daripada penjadwalan manual. Hasil penjadwalan dengan algoritma *graph colouring* dapat dilihat pada tabel 6.5. Tabel tersebut menggambarkan penjadwalan mata kuliah di setiap sesi dan hari. Angka dalam tabel merepresentasikan kode mata kuliah sesuai dengan yang ada pada tabel 4.5.

Tabel 6.5 Hasil Penjadwalan Ujian Menggunakan Graph Colouring

Session/Day	Day 1	Day 2	Day 3	Day 4	Day 5
session 1	28 , 32	24 , 29 , 34	22 , 23 , 39	0 , 2	1
session 2	4 , 12 , 25 , 36	7 , 17 , 21	31 , 37 , 40	19 , 35 , 38	9 , 14
session 3	20 , 30	8 , 13 , 15	3 , 10 , 16 , 27	11 , 18 , 26 , 33	5 , 6

6.5. Hasil Uji Coba Algoritma *Great Deluge*

Algoritma *great deluge* digunakan untuk melakukan optimasi pada jadwal yang telah di inisiasi sebelumnya. Beberapa parameter yang diperlukan dalam menjalankan algoritma *great deluge* dapat dilihat pada gambar 6.5.

Tabel 6.6 Parameter *great deluge*

Level	<i>Proximity initial solution</i>
Iterasi	1000 - 1000000
Nilai yang diinginkan (D)	30

LLH	Random move dan random swap
------------	-----------------------------

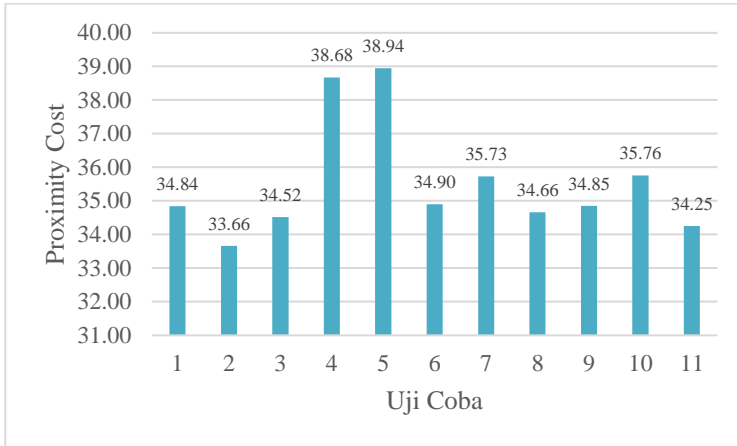
Nilai parameter level diiniasi sama dengan nilai *proximity cost* jadwal awal. Jumlah iterasi, nilai D dan *low level heuristic* mengacu pada penelitian[1]. Uji coba dilakukan sebanyak sebelas kali dikarenakan algoritma *great deluge* merupakan algoritma yang bersifat stokastik sehingga tidak dapat menghasilkan solusi pasti karena melibatkan probabilitas.

6.5.1. Skenario Uji Coba 1 : Perubahan Iterasi

Pada skenario uji coba pertama dilakukan pengubahan jumlah iterasi untuk mengetahui apakah jumlah iterasi mempengaruhi fungsi tujuan akhir. Dalam penelitian ini penulis melakukan uji coba dengan mengubah nilai iterasi menjadi 1000, 10000, 100000 dan 1000000. Setiap satu nilai iterasi dilakukan uji coba sebanyak sebelas kali. Pada skenario uji coba satu menggunakan *low level heuristic* random move dan swap.

6.5.1.1. Iterasi 1000

Pada uji coba ini nilai parameter *I* diubah menjadi 1000 yang berarti algoritma melakukan iterasi sebanyak 1000 kali. Uji coba ini dilakukan sebanyak sebelas kali.



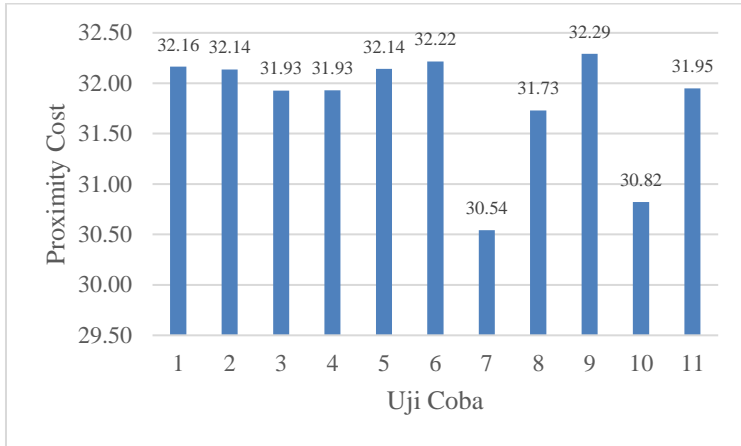
Gambar 6.3 Grafik Proximity Cost Untuk Iterasi 1000

Grafik hasil uji coba skenario satu dengan nilai iterasi 1000 dapat dilihat pada gambar 6.3. Dari hasil tersebut didapatkan nilai fungsi tujuan sebesar 33.66 sebagai nilai fungsi tujuan terbaik dan 35.53 sebagai rata-rata nilai fungsi tujuan uji coba ini.

6.5.1.2. Iterasi 10000

Pada uji coba ini algoritma melakukan iterasi sebanyak 10000 kali untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan. Hasil *proximity cost* dengan melakukan iterasi 10000 dapat dilihat pada gambar 6.3.

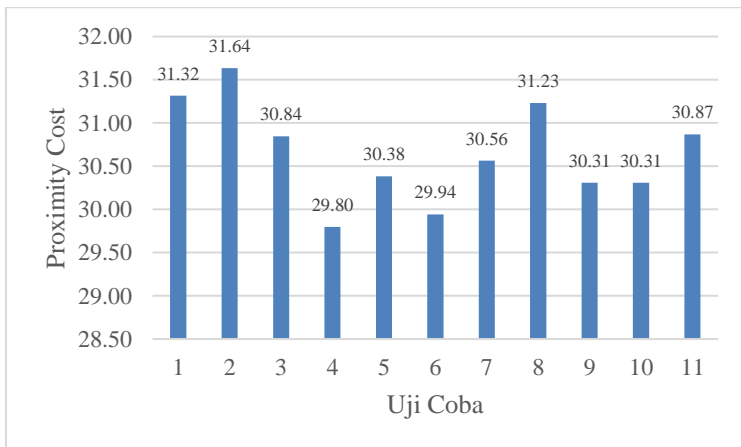
Dari sebelas kali percobaan didapatkan hasil dengan nilai fungsi tujuan terbaik yaitu 30.54 pada percobaan ke-empat sedangkan untuk rata-rata uji coba ini yaitu sebesar 31.81. Nilai fungsi tujuan yang didapat pada percobaan ini lebih baik dibandingkan dengan uji coba dengan jumlah iterasi 1000.



Gambar 6.4 Grafik Proximity Cost Untuk Iterasi 10000

6.5.1.3. Iterasi 100000

Pada uji coba ini nilai parameter I diubah menjadi 100000 untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.



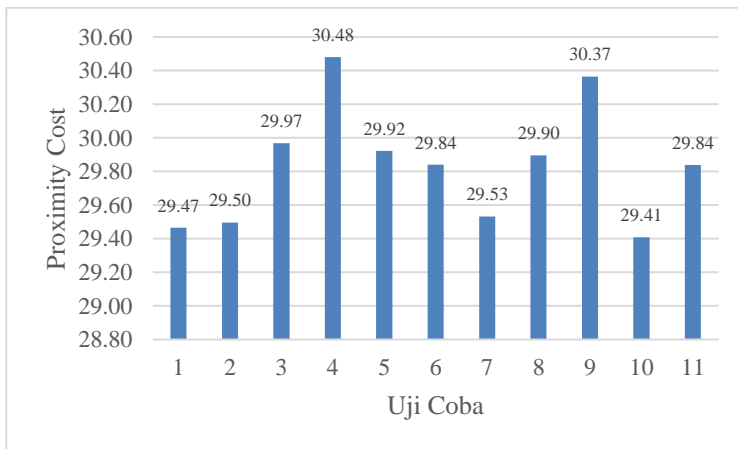
Gambar 6.5 Grafik Proximity Cost Untuk Iterasi 100000

Hasil yang didapatkan pada percobaan ini yaitu 29.80 sebagai nilai *proximity cost* yang paling rendah dan rata-rata nilai fungsi tujuan percobaan uji coba ini adalah 30.67. Hasil ini

menunjukkan bahwa uji coba algoritma dengan iterasi 100000 lebih baik dibanding dengan iterasi 1000 dan 10000.

6.5.1.4. Iterasi 1000000

Pada uji coba ini algoritma melakukan iterasi sebanyak 1000000 kali dengan dilakukan sebelas kali percobaan untuk melihat pengaruhnya terhadap fungsi tujuan serta jadwal yang dihasilkan.



Gambar 6.6 Grafik Proximity Cost Untuk Iterasi 1000000

Dari hasil tersebut didapatkan nilai fungsi tujuan sebesar 29.41 sebagai nilai *proximity cost* terbaik dan 29.84 sebagai rata-rata nilai fungsi tujuan uji coba ini. Rata-rata tersebut menunjukkan bahwa iterasi 1000000 lebih baik dibandingkan dengan iterasi sebelumnya.

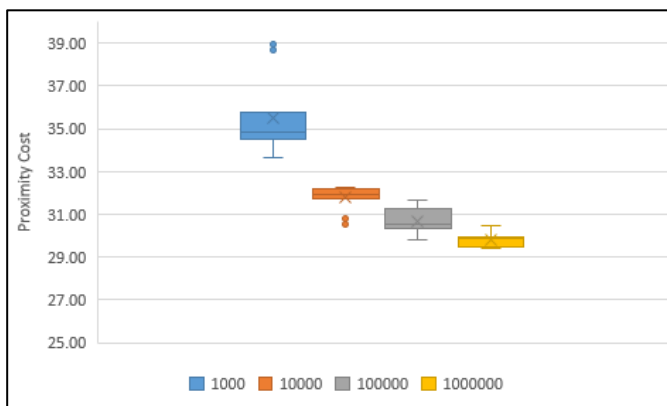
6.5.1.5. Perbandingan Hasil Uji Coba Skenario 1

Hasil uji coba skenario 1 dengan melakukan pengubahan jumlah iterasi 1000, 10000, 100000 dan 1000000 dapat dilihat pada tabel 6.7.

Tabel 6.7 Perbandingan Hasil Uji Coba Skenario 1

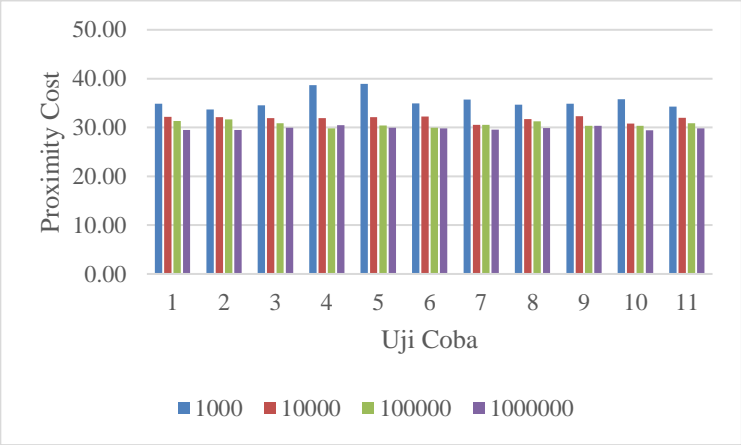
Uji Coba	1000	10000	100000	1000000
1	34.84	32.16	31.32	29.47
2	33.66	32.14	31.64	29.50
3	34.52	31.93	30.84	29.97
4	38.68	31.93	29.80	30.48
5	38.95	32.14	30.38	29.92
6	34.90	32.22	29.94	29.84
7	35.73	30.54	30.57	29.53
8	34.66	31.73	31.23	29.90
9	34.85	32.29	30.31	30.37
10	35.76	30.82	30.31	29.41
11	34.25	31.95	30.87	29.84
MAX	38.95	32.29	31.64	30.48
MIN	33.66	30.54	29.80	29.41
AVE	35.53	31.81	30.66	29.84

Dari hasil percobaan skenario 1 didapatkan bahwa iterasi dengan jumlah 1000000 menghasilkan nilai rata-rata yang paling baik. *Proximity cost* dengan nilai 29.41 merupakan nilai *proximity cost* terbaik dari 44 kali percobaan uji skenario 1. Dapat disimpulkan bahwa jumlah iterasi didalam algoritma *great deluge* mempengaruhi nilai fungsi tujuan akhir, semakin



Gambar 6.7 Grafik Box Plot Skenario 1

banyak jumlah iterasi maka hasil yang didapatkan juga semakin baik. Pada gambar 6.5 menunjukkan bahwa jumlah iterasi pada algoritma *great deluge* dapat mempengaruhi nilai fungsi tujuan akhir



Gambar 6.8 Grafik Perbandingan Hasil Uji Coba Skenario 1

Hasil jadwal dengan fungsi tujuan terbaik dari setiap iterasi pada skenario 1 dapat dilihat pada tabel 6.8.

Tabel 6.8 Hasil Jadwal Fungsi Tujuan Terbaik Pada Skenario 1

Jumlah Iterasi	Fungsi Tujuan Terbaik	Usulan Jadwal
1000	33.664	10 13 10 9 8 15 15 15 15 14 9 12 2 6 14 6 12 2 12 14 4 5 7 7 11 15 14 9 1 4 3 8 1 12 4 11 2 8 13 13 8
10000	30.543	10 13 15 9 2 9 15 10 15 14 13 12 9 10 14 6 1 15 12 11 13 5 7 9 5 1 9 14 1 4 3 8 4 12 15 11 2 8 11 7 8
100000	29.798	10 14 11 9 8 9 11 15 2 7 13 12 9 10 7 6 14 6 12 15 4 5 15 10 1 15 10 7 1 4 3 8 9 12 2 6 2 8 13 14 8

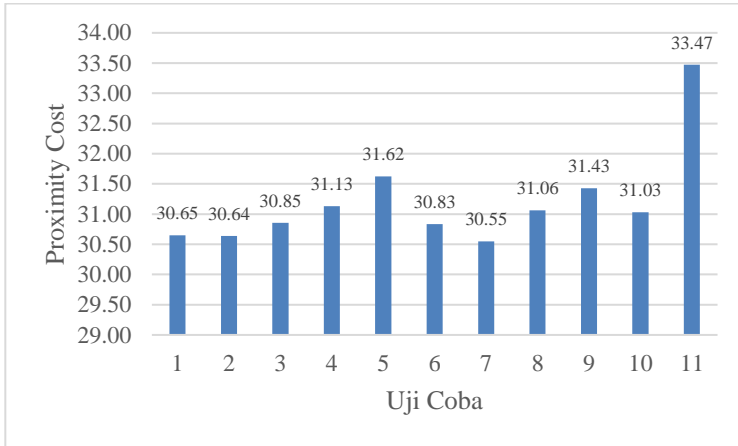
1000000	29.41	3 10 10 7 13 14 14 15 12 13 7 4 14 3 13 12 8 11 4 6 1 5 15 5 2 15 14 7 9 11 1 13 4 4 15 2 6 8 2 10 3
---------	-------	--

6.5.2. Skenario Uji Coba 2 : Pemilihan *Low Level Heuristic*

Pada penelitian ini implementasi algoritma *great deluge* menggunakan 2 metode *low level heuristic* yaitu metode *move* dengan cara mengubah slot waktu suatu mata kuliah ke timeslot lain secara acak dan metode *swap* dengan cara memilih dua mata kuliah secara acak lalu menukar slot mata kuliah tersebut. Skenario uji coba 2 dilakukan dengan mengubah pemilihan metode *low level heuristic* yaitu metode *move* saja, metode *swap* saja dan *random* yaitu menggabungkan metode *move* dan metode *swap*. Dalam percobaan ini dilakukan uji coba sebanyak masing-masing 11 kali.

6.5.2.1. Low Level Heuristic : Move

Pada uji coba ini jadwal dioptimasi dengan menggunakan metode *move* saja. Percobaan dilakukan sebanyak 11 kali dengan jumlah iterasi sebanyak 10000.

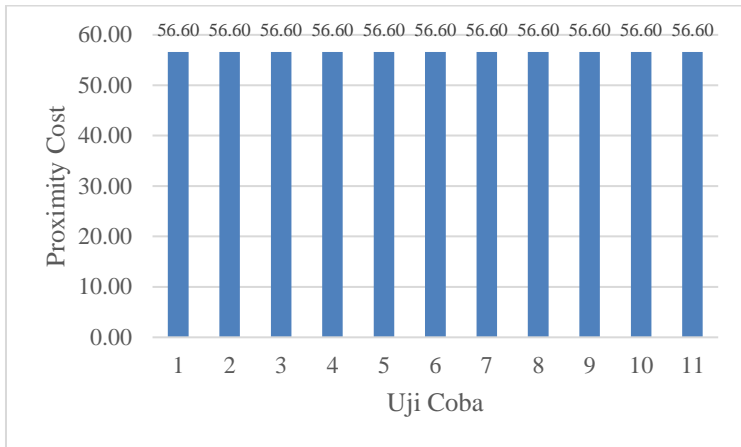


Gambar 6.9 Grafik Proximity Cost LLH Move

Dari 11 kali hasil percobaan didapatkan nilai *proximity cost* sebesar 30.55 sebagai nilai fungsi tujuan terbaik. Sedangkan rata-rata terbaik dalam menggunakan *low level heuristic move* yaitu 31.20.

6.5.2.2. Low Level Heuristic : Swap

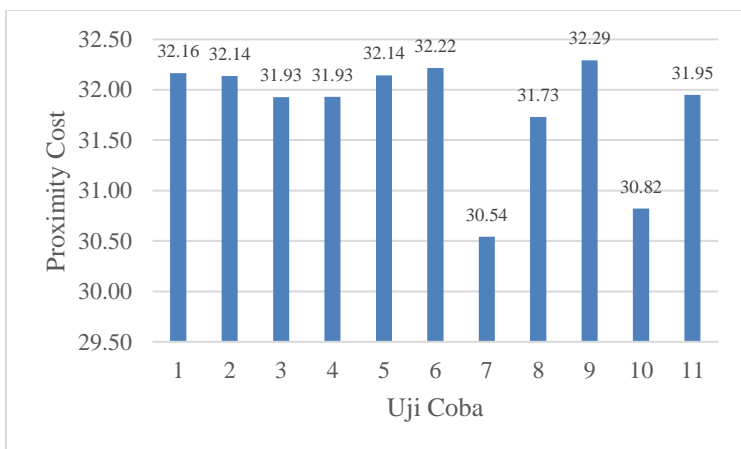
Pada uji coba ini jadwal dioptimasi dengan menggunakan metode *swap* saja. Percobaan dilakukan sebanyak 11 kali dengan jumlah iterasi 10000. Dari hasil uji coba didapatkan bahwa tidak ada perubahan nilai *proximity cost* karena jadwal yang dioptimasi dengan metode *swap* tidak berubah. Sehingga jadwal yang dihasilkan pada uji coba ini sama dengan jadwal solusi awal.



Gambar 6.10 Grafik Proximity Cost LLH Swap

6.5.2.3. Low Level Heuristic : Move dan Swap

Pada uji coba ini jadwal dioptimasi dengan menggunakan metode *move* saja. Percobaan dilakukan sebanyak 11 kali dengan jumlah iterasi yang berbeda-beda.



Gambar 6.11 Grafik Proximity Cost LLH Move dan Swap

Dari hasil uji coba menggunakan gabungan metode *move* dan *swap* didapatkan nilai fungsi tujuan terbaik dengan *proximity*

cost sebesar 30.54. Sedangkan rata-rata terbaik dalam melakukan optimasi jadwal yaitu sebesar 31.80

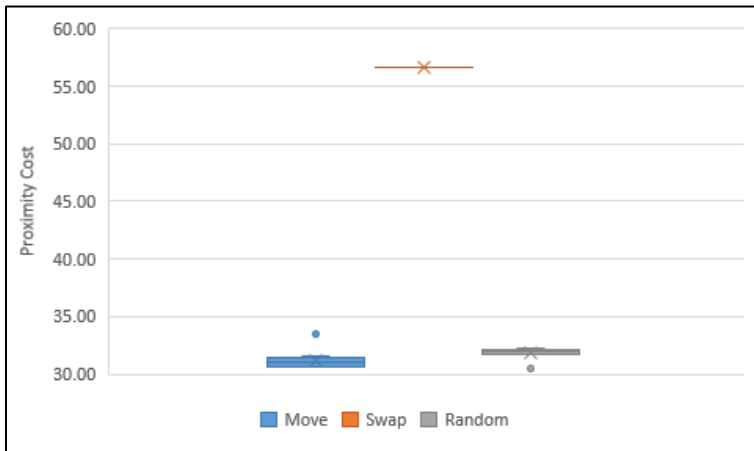
6.5.2.4. Perbandingan hasil Uji Coba Skenario 2

Iterasi yang digunakan dalam membandingkan uji skenario 2 yaitu menggunakan iterasi 10000 dan masing-masing dilakukan percobaan sebanyak 11 kali.

Tabel 6.9 perbandingan hasil percobaan skenario 2

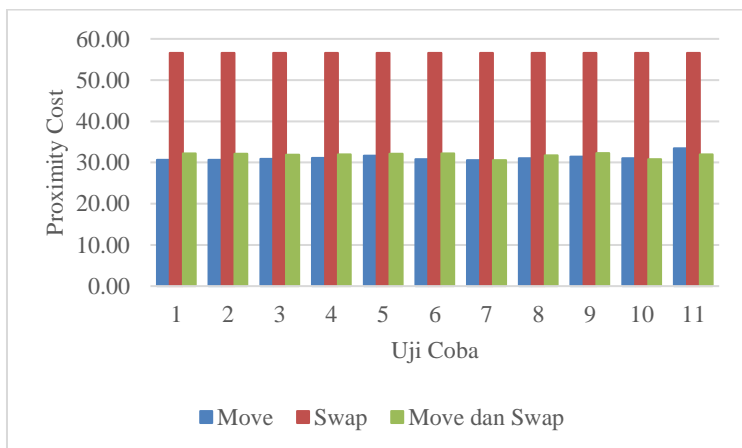
Percobaan	Move	Swap	Random
1	30.65	56.60	32.16
2	30.64	56.60	32.14
3	30.85	56.60	31.93
4	31.13	56.60	31.93
5	31.62	56.60	32.14
6	30.83	56.60	32.22
7	30.55	56.60	30.54
8	31.06	56.60	31.73
9	31.43	56.60	32.29
10	31.03	56.60	30.82
11	33.47	56.60	31.95
MAX	33.47	56.60	32.29
MIN	30.55	56.60	30.54
AVE	31.20	56.60	31.81

Dari hasil percobaan yang telah dilakukan menunjukkan bahwa metode *move* menghasilkan fungsi tujuan yang lebih baik dibanding dengan metode *swap* dan *random*. Pada gambar 6.11 menunjukkan bahwa metode *random* lebih stabil dibanding metode *move* dan *swap*.



Gambar 6.12 Grafik Box Plot Skenario 2

Perbandingan *proximity cost* pada skenario 2 dapat dilihat pada gambar grafik berikut.



Gambar 6.13 Grafik Perbandingan Hasil Uji Coba Skenario 2

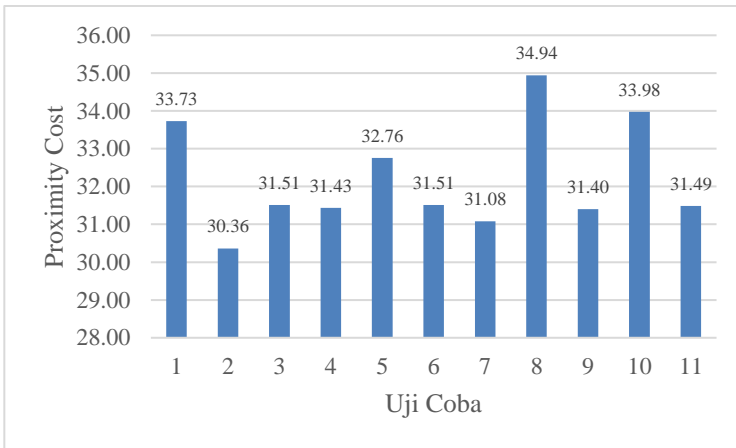
6.5.3. Skenario Uji Coba 3 : Perubahan Waktu

Pada skenario uji coba ketiga dilakukan pengoptimasian dengan mengubah waktu eksekusi. Pada pengujian ini algoritma akan terus dijalankan sampai batas waktu tertentu. Hasil akhir uji

coba skenario 3 merupakan hasil yang didapatkan pada saat akhir waktu yang telah ditentukan.

6.5.3.1. Perubahan waktu 1 menit

Pada uji coba ini jadwal dioptimasi dengan menggunakan waktu eksekusi program selama 1 menit. Percobaan dilakukan sebanyak 11 kali.

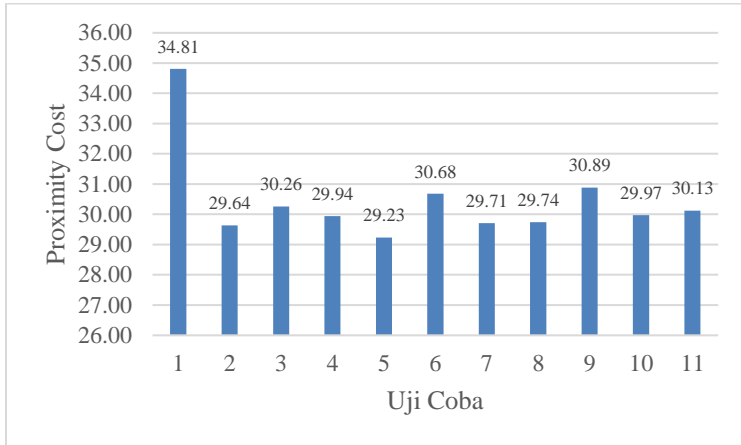


Gambar 6.14 Grafik Proximity Cost Dengan Waktu Eksekusi 1 Menit

Dari 11 kali hasil percobaan didapatkan nilai *proximity cost* sebesar 30.36 sebagai nilai fungsi tujuan terbaik. Sedangkan rata-rata terbaik fungsi tujuan akhirs dengan waktu eksekusi 1 menit sebesar 32.20.

6.5.3.2. Perubahan waktu 5 menit

Pada uji coba ini dilakukan optimasi jadwal dengan menggunakan waktu eksekusi selama 5 menit. Percobaan dilakukan sebanyak 11 kali.

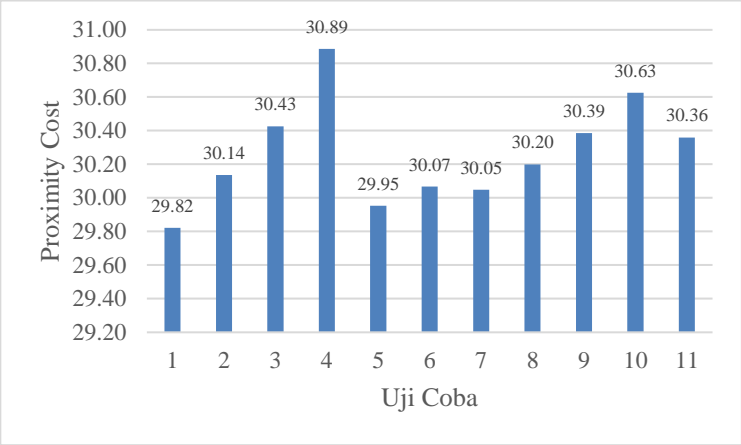


Gambar 6.15 Grafik Proximity Cost Dengan Waktu Eksekusi 5 Menit

Dari hasil uji coba menjalankan algoritma *great deluge* menggunakan waktu eksekusi 5 menit menghasilkan nilai fungsi tujuan terbaik dengan *proximity cost* sebesar 29.33. Sedangkan rata-rata terbaik dalam melakukan optimasi jadwal ujian yaitu sebesar 30.45.

6.5.3.3. Perubahan waktu 10 menit

Dalam percobaan ini dilakukan uji coba sebanyak 11 kali dengan waktu eksekusi selama 10 menit. Nilai rata-rata *proximity* yang didapatkan dari hasil percobaan dengan waktu eksekusi selama 10 menit yaitu 30.26. Dari rata-rata tersebut menunjukkan bahwa iterasi dengan waktu eksekusi 10 menit lebih baik dibanding dengan waktu eksekusi 1 menit dan 5 menit. Proximity terbaik pada percobaan ini yaitu 29.82



Gambar 6.16 Grafik Proximity Cost Dengan Waktu Eksekusi 10 Menit

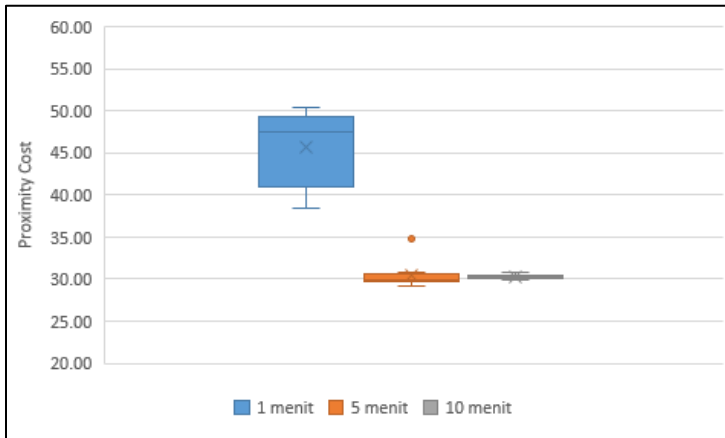
6.5.3.4. Perbandingan Hasil Uji Coba Skenario 3

Dari beberapa hasil percobaan dengan perubahan waktu eksekusi 5 menit, 10 menit, dan 30 menit. Masing-masing dilakukan percobaan sebanyak 11 kali.

Tabel 6.10 Perbandingan Hasil Percobaan Skenario 3

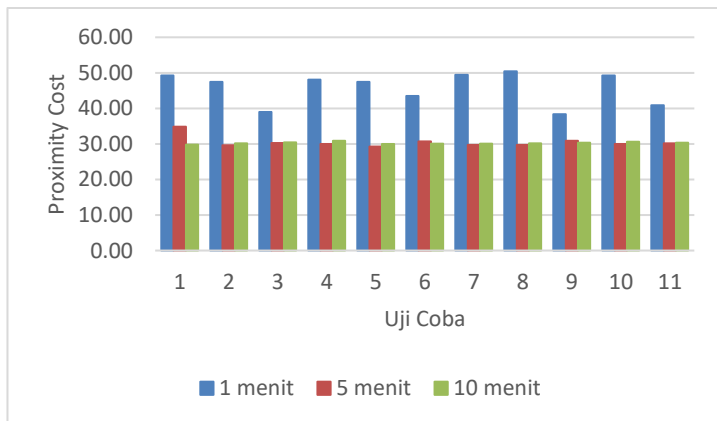
Percobaan	1 mnt	5 mnt	10 mnt
1	33.73	34.81	29.82
2	30.36	29.64	30.14
3	31.51	30.26	30.43
4	31.43	29.94	30.89
5	32.76	29.23	29.95
6	31.51	30.68	30.07
7	31.08	29.71	30.05
8	34.94	29.74	30.20
9	31.40	30.89	30.39
10	33.98	29.97	30.63
11	31.49	30.13	30.36
MAX	34.94	34.81	30.89
MIN	30.36	29.23	29.82
AVG	32.20	30.45	30.26

Pada percobaan diatas dapat dilihat bahwa semakin lama waktu eksekusi program maka semakin baik pula nilai fungsi tujuan akhir yang dihasilkan serta data yang dihasilkan lebih stabil.



Gambar 6.17 Grafik Box Plot Skenario 3

Perbandingan *proximity cost* pada skenario 2 dapat dilihat pada gambar grafik berikut.



Gambar 6.18 Grafik Perbandingan Hasil Uji Coba Skenario 3

6.5.4. Pembahasan Hasil Skenario Uji Coba

1. Pemilihan jumlah iterasi pada pengoptimasian jadwal dengan menggunakan algoritma *great deluge* memiliki pengaruh terhadap nilai fungsi tujuan akhir, semakin banyak jumlah iterasi maka hasil fungsi tujuan akhir yang didapatkan semakin baik.
2. Pemilihan *low level heuristic* memiliki pengaruh dalam melakukan optimasi jadwal. Dari uji coba yang dilakukan method *move* lebih baik dibandingkan dengan method *swap* dan *random move* dan *swap*.
3. Pemilihan waktu eksekusi program dijalankan pada optimasi *great deluge* dapat mempengaruhi hasil nilai fungsi tujuan, semakin lama waktu eksekusi semakin baik nilai fungsi tujuan yang dihasilkan.

6.6. Perbandingan dengan Algoritma Lain

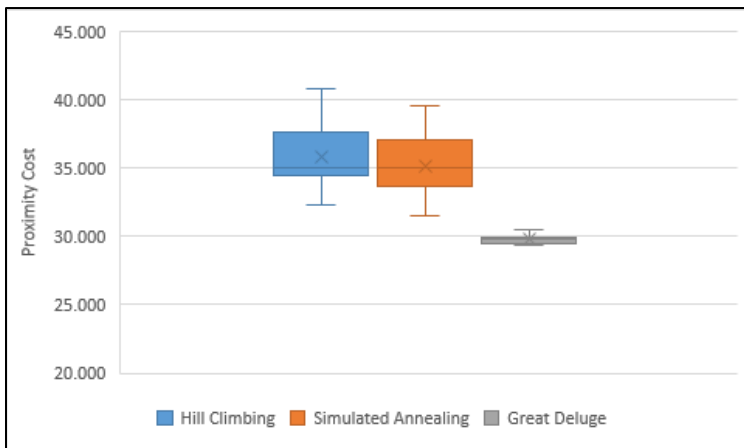
Pada uji coba ini algoritma *great deluge* akan dibandingkan dengan algoritma optimasi lain yaitu *hill climbing* dan *simulated annealing*. Percobaan dilakukan sebanyak 11 kali dengan jumlah iterasi sebanyak 1000000 iterasi. Uji coba ini dilakukan untuk menentukan perbandingan kinerja algoritma dalam menentukan solusi terbaik dalam jumlah iterasi tertentu.

Tabel 6.11 Perbandingan Algoritma Optimasi

Percobaan	Hill Climbing	Simulated Annealing	Great Deluge
1	32.27	31.49	29.47
2	35.99	33.94	29.50
3	33.53	33.65	29.97
4	37.98	34.46	30.48
5	34.50	37.15	29.92
6	34.54	37.04	29.84
7	37.69	33.25	29.53
8	40.84	39.59	29.90
9	37.09	36.33	30.37
10	34.98	34.98	29.41

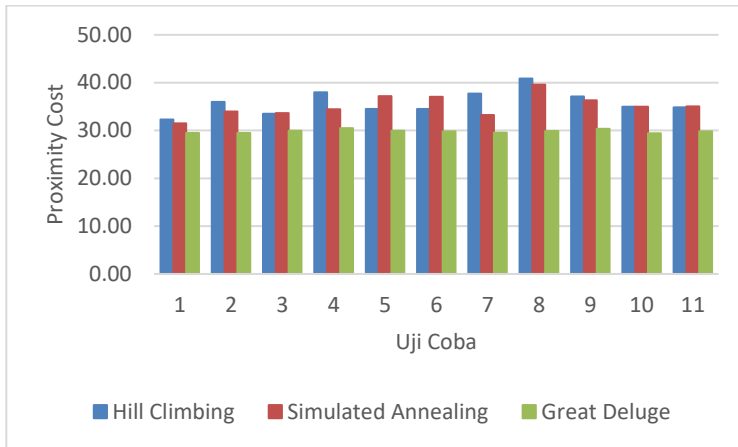
11	34.82	35.07	29.84
MAX	40.84	39.59	30.48
MIN	32.27	31.49	29.41
AVE	35.84	35.18	29.84

Dari hasil percobaan diatas dapat disimpulkan bahwa algoritma *great deluge* memiliki nilai fungsi tujuan akhir terbaik dibandingkan dengan algoritma optimasi lainnya. Sedangkan algoritma *simulated annealing* lebih baik dibanding dengan algoritma *hill climbing*. Pada gambar 6.13 grafik blox pot menunjukkan bahwa algoritma great deluge memiliki penyebaran data yang kecil dan lebih stabil dibanding dengan algoritma lainnya. Hasil penjadwalan ujian menggunakan algoritma *great deluge* dapat dilihat pada Lampiran C dan lampiran E.



Gambar 6.19 Grafik Box Plot Perbandingan Algoritma Optimasi

Perbandingan *proximity cost* yang dihasilkan pada setiap algoritma optimasi dapat dilihat pada gambar grafik berikut.



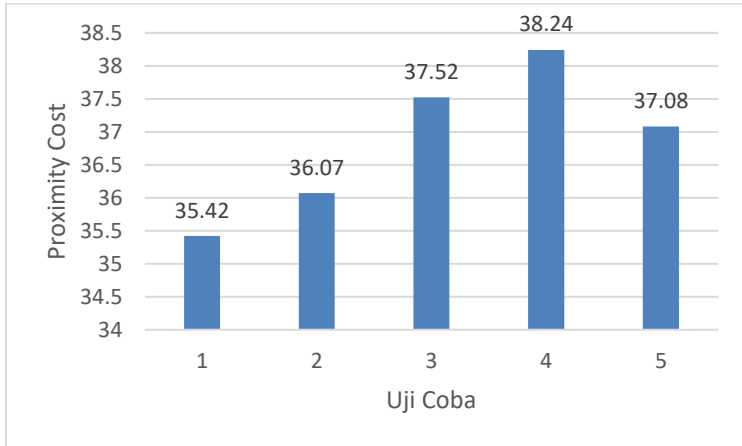
Gambar 6.20 Grafik Perbandingan Algoritma Optimasi

6.7. Penjadwalan Ujian Departemen Sistem Informasi

Data masukan yang digunakan untuk menjadwalkan ujian pada Departemen Sistem Informasi didapatkan dari penelitian [1]. Didalam tugas akhir ini dilakukan uji coba penjadwalan ujian pada Departemen Sistem Informasi menggunakan algoritma *great deluge* yang akan dibandingkan dengan algoritma pada penelitian sebelumnya [1] yaitu algoritma simulated annealing.

6.7.1. Algoritma *Great Deluge*

Dalam percobaan ini, penjadwalan ujian pada Departemen Sistem Informasi dilakukan percobaan sebanyak lima kali dan setiap percobaan dilakukan iterasi sebanyak 1000000 kali dengan menggunakan metode *swap* dan *move*.

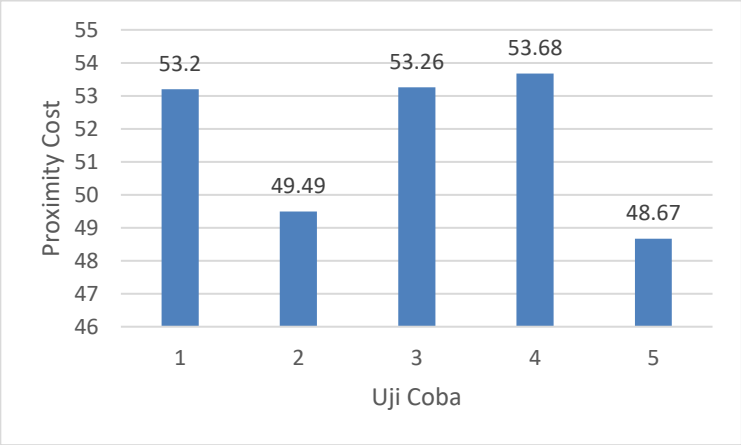


Gambar 6.21 Grafik Proximity Cost Algoritma Great Deluge Pada Departemen Sistem Informasi

Dari gambar diatas menunjukkan bahwa *proximity cost* terbaik pada percobaan ini yaitu 35.42 dengan rata-rata nilai *proximity cost* sebesar 38.87.

6.7.2. Algoritma Simulated Annealing

Uji Coba yang dilakukan dalam penjadwalan ujian dengan menggunakan algoritma *simulated annealing* dilakukan sebanyak 1000000 iterasi dengan masing-masing lima kali percobaan.



Gambar 6.22 Grafik Proximity Cost Algoritma Simulated Annealing Pada Departemen Sistem Informasi

Dari hasil percobaan penjadwalan ujian menggunakan algoritma *simulated annealing* pada Departemen Sistem Informasi didapatkan hasil nilai *proximity* terbaik yaitu 35.42 dengan rata-rata nilai *proximity* sebesar 51.66. Hal ini menunjukkan bahwa penjadwalan ujian dengan algoritma *simulated annealing* pada Departemen Sistem Informasi tidak lebih baik dibandingkan dengan penjadwalan menggunakan algoritma *great deluge*.

6.7.3. Perbandingan Algoritma Great Deluge dan Simulated Annealing

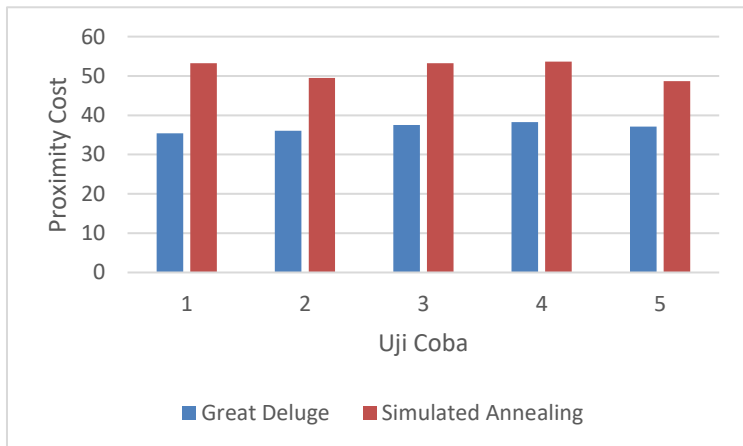
Hasil perbandingan percobaan penjadwalan ujian menggunakan algoritma *great deluge* dan algoritma *simulated annealing* dapat dilihat pada tabel berikut.

Tabel 6.12 Perbandingan Proximity Algoritma Great Deluge dan Simulated Annealing

Percobaan	Great Deluge	Simulated Annealing
1	35.42	53.2

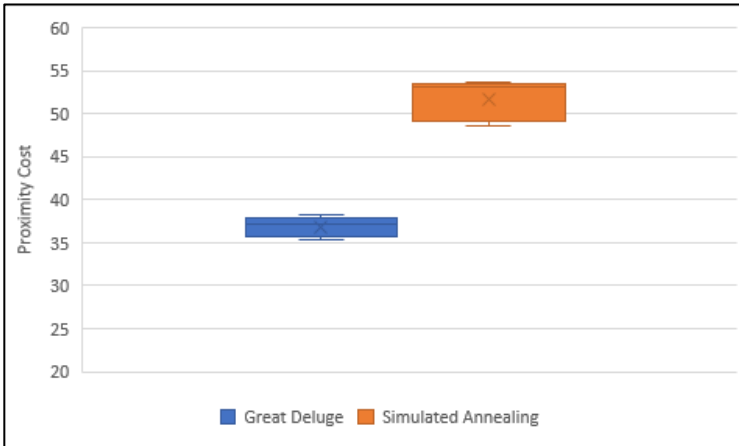
Percobaan	Great Deluge	Simulated Annealing
2	36.07	49.49
3	37.52	53.26
4	38.24	53.68
5	37.08	48.67
MAX	38.24	53.68
MIN	35.42	48.67
AVG	36.87	51.66

Dari hasil percobaan diatas dapat disimpulkan bahwa algoritma *great deluge* memiliki nilai fungsi tujuan akhir terbaik dibanding dengan algoritma *simulated annealing* yaitu dengan nilai rata-rata *proximity cost* sebesar 36.87 dan menghasilkan nilai *proximity* terbaik sebesar 35.42. Perbandingan *proximity cost* yang dihasilkan pada setiap algoritma optimasi dapat dilihat pada gambar grafik berikut.



Gambar 6.23 Grafik Perbandingan Proximity Algoritma Great Deluge dan Simulated Annealing

Pada gambar 6.24 menunjukkan hasil box plot dari perbandingan hasil dari optimasi penjadwalan ujian menggunakan algoritma great deluge dan simulated annealing.



Gambar 6.24 Grafik Box Plot Perbandingan Algoritma Great Deluge dan Simulated Annealing

6.8. Penjadwalan Ruang Ujian

Pada tugas akhir ini penjadwalan ruang ujian di asumsikan bahwa semua mata kuliah yang di ujikan memiliki durasi waktu yang sama. Dua mata kuliah atau lebih dapat dijadwalkan dalam satu ruang ujian apabila kapasitas ruang ujian masih mencukupi.

6.8.1. Penjadwalan Ruang Ujian Hasil Initial Solution

Penjadwalan ruang ujian hasil dari *initial solution* dapat dilihat pada tabel 6.14, untuk potongan tampilan di aplikasi dapat dilihat pada tabel 6.13. Tabel tersebut menggambarkan penjadwalan mata kuliah di setiap sesi dan ruang ujian. Angka dalam tabel merepresentasikan kode mata kuliah sesuai dengan yang ada pada tabel 4.5. Untuk hasil penjadwalan waktu dan ruang ujian initial solution dapat dilihat pada lampiran B.

Tabel 6.13 Potongan Tampilan Hasil Penjadwalan Ruang Pada Jadwal Initial Solution

Session	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106
timeslot 1	28	28	28	28 , 32	32	32
timeslot 2	4 , 12 , 25	25	25	25	25	25 , 36
timeslot 3	20	20	20	20	20 , 30	30
timeslot 4	24	24	24	24	24 , 29	29

Tabel 6.14 Hasil Penjadwalan Ruang Pada Jadwal Initial Solution

Sesi	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas
TS 1	28	28	28	28 , 32	32	32	32		
TS 2	4 , 12 , 25	25	25	25	25	25 , 36	36	36	36
TS 3	20	20	20	20	20 , 30	30	30	30	
TS 4	24	24	24	24	24 , 29	29	29	29 , 34	34
TS 5	7	7	7	7	7 , 17	17	17	17	17 , 21
TS 6	8 , 13 , 15	15	15	15	15	15			
TS 7	22	22	22	22 , 23	23	23	23	23 , 39	
TS 8	31 , 37 , 40	40	40	40					
TS 9	3 , 10	10 , 16 , 27	27	27	27				
TS 10	0 , 2	2	2	2	2				

Sesi	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas
TS 11	19 , 35 , 38	38	38	38					
TS 12	11 , 18 , 26	26	26	26 , 33	33				
TS 13	1								
TS 14	9 , 14								
TS 15	5 , 6	6							

6.8.2. Penjadwalan Ruang Ujian Hasil Algoritma Great Deluge

Penjadwalan ruang ujian hasil dari optimasi algoritma *great deluge* dengan nilai *proximity* yang paling baik yaitu 29.41 dapat dilihat pada tabel 6.16, untuk potongan tampilan di aplikasi dapat dilihat pada tabel 6.15 . Tabel tersebut menggambarkan penjadwalan mata kuliah di setiap sesi dan ruang ujian. Angka dalam tabel merepresentasikan kode mata kuliah sesuai dengan yang ada pada tabel 4.5. Untuk hasil penjadwalan waktu dan ruang ujian initial solution dapat dilihat pada lampiran A.

Tabel 6.15 Potongan Tampilan Hasil Penjadwalan Ruang Pada Jadwal Hasil Optimasi Great Deluge

Session	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106
timeslot 1	20	20	20	20	20 , 30	30
timeslot 2	24	24	24	24	24 , 35 , 38	38
timeslot 3	0 , 13	13 , 40	40	40		
timeslot 4	11 , 18 , 32	32	32	32 , 33		
timeslot 5	21	21	21	21 , 23	23	23

Tabel 6.16 Hasil Penjadwalan Ruang Pada Jadwal Hasil Optimasi Great Deluge

Sesi	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas
TS 1	20	20	20	20	20 , 30	30	30	30	
TS 2	24	24	24	24	24 , 35 , 38	38	38	38	
TS 3	0 , 13	13 , 40	40	40					
TS 4	11 , 18 , 32	32	32	32 , 33					
TS 5	21	21	21	21 , 23	23	23	23	23	
TS 6	19 , 36	36	36	36	36				
TS 7	3 , 10	10 , 27	27	27	27				
TS 8	16 , 37	37							

Sesi	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas
TS 9	28	28	28	28					
TS 10	1, 2	2	2	2	2, 39				
TS 11	17	17	17	17	17, 29	29	29	29	
TS 12	8, 15	15	15	15	15				
TS 13	4, 9, 14	14, 31							
TS 14	5, 6	6, 12, 26	26	26	26				
TS 15	7	7	7	7	7, 22	22	22	22, 25	25, 34

BAB VII

SARAN DAN KESIMPULAN

7.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa :

1. Algoritma *graph colouring - great deluge* dapat digunakan untuk membuat jadwal ujian yang lebih optimal di Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember dengan menggunakan 15 slot waktu ujian.
2. Dari perbandingan uji coba yang telah dilakukan diketahui bahwa algoritma *great deluge* dengan *proximity cost* rata-rata 29.41 lebih baik dibanding dengan penjadwalan manual, algoritma *hill climbing* dan algoritma *simulated annealing* dengan nilai *proximity cost* berturut-turut 51.15, 35.18 dan 35.84.
3. Optimasi penjadwalan waktu dan ruang ujian otomatis menggunakan algoritma *graph colouring - great deluge* memudahkan dalam melakukan perubahan jadwal.

7.2. Saran

Saran yang diberikan berdasarkan hasil pengerjaan tugas akhir untuk penelitian selanjutnya antara lain :

1. Pada pengerjaan tugas akhir ini, dalam implementasi algoritma *graph colouring* hanya menggunakan satu pendekatan saja, yaitu *Largest Degree*. Sehingga pada penelitian selanjutnya diharapkan dapat menggunakan pendekatan pengurutan yang lain yaitu *Largest weighted degree*, *Largest enrollment* dan *Saturation degree*.
2. Penelitian selanjutnya dapat menggunakan *soft constraint* ruang dimana dua mata kuliah atau lebih

yang memiliki durasi waktu ujian yang berbeda dapat dijadwalkan pada ruang yang berbeda.

3. Pada pemilihan *low level heuristic* bisa ditambahkan dengan metode lain seperti metode *random move two* atau *kempe chain*, untuk mengetahui pengaruh kinerja algoritma serta hasil akhir nilai fungsi tujuan.
4. Membuat visualisasi jadwal setiap mahasiswa, sehingga memudahkan mahasiswa dalam melihat jadwal ujian.
5. Menambahkan fitur *download* agar jadwal ujian yang dihasilkan dapat disimpan dalam bentuk pdf untuk kemudian dapat dicetak.
6. Menambahkan penjadwalan pengawas ujian setiap ruang ujian.

DAFTAR PUSTAKA

- [1] D. Kusumawardani and A. Mukhlason, “Optimasi Penjadwalan Ujian Otomatis Dengan Menggunakan Algoritma Greedy – Simulated Annealing – Hyper Heuristic (Studi Kasus: Departemen Sistem Informasi, ITS),” pp. 1–7.
- [2] A. Mukhlason, “Solver Penjadwal Ujian Otomatis Dengan Algoritma Maximal Clique dan Hyper-heuristics,” pp. 18–19, 2017.
- [3] A. K. Mandal and M. N. M. Kahar, “Solving examination timetabling problem using partial exam assignment with hill climbing search,” *ISCAIE 2015 - 2015 IEEE Symp. Comput. Appl. Ind. Electron.*, no. I4ct, pp. 84–89, 2015.
- [4] S. Abdullah, “HEURISTIC APPROACHES FOR UNIVERSITY TIMETABLING PROBLEMS,” no. June, 2006.
- [5] A. Wren, “Scheduling, timetabling and rostering – A special relationship? The Practice and Theory of Automated Timetabling,” *PATAT I*, pp. 46–75, 1996.
- [6] E. K. Burke and S. Petrovic, “Recent Research Directions in Automated Timetabling Edmund Kieran Burke and Sanja Petrovic,” *Eur. J. Oper. Res.*, vol. 140, no. 2, pp. 266–280, 2002.
- [7] M.W. Carter and G. Laporte, “Recent developments in practical examination timetabling. The Practice and Theory of Automated Timetabling I,” pp. 3–21, 1996.
- [8] G. L. and J. W. C. M.W. Carter, “A general examination scheduling system,” pp. 109–120.
- [9] P. H. F. and R. F. W. E.K. Burke, D.G. Elliman, “Examination timetabling in British universities,” pp. 76–92.
- [10] T. B. Cooper dan J. H. Kingston, “The complexity of timetable construction problems. In Edmund Burke and Peter Ross, editors, Practice and Theory of Automated Timetabling,” *Springer Berlin Heidelb.*, vol. 1153, pp.

- 281–295, 1996.
- [11] A. Muklason, A. J. Parkes, E. Özcan, B. Mccollum, and P. McMullan, “Fairness in examination timetabling: Student preferences and extended formulations,” *Appl. Soft Comput. J.*, vol. 55, pp. 302–318, 2017.
 - [12] B. Mccollum, P. J. McMullan, A. J. Parkes, E. K. Burke, and S. Abdullah, “An Extended Great Deluge Approach to the Examination Timetabling Problem,” 2009.
 - [13] R. Munir, “Algoritma Greedy,” ITB, 2004.
 - [14] R. Hartanto, “Algoritma Greedy dalam Strategi Permainan Centipede,” no. 13515107, 2017.
 - [15] S. Burke, Edmund & Kendall, Graham & Newall, Jim & Hart, Emma & Ross, Peter & Schulenburg, “Hyper-Heuristics: An Emerging Direction in Modern Search Technology,” *Handb. Metaheuristics*, no. July, 2003.
 - [16] S. D. O. Ilboudo, I. Sombi??, A. K. Soubeiga, and T. Dr??bel, “Algorithm for knapsack problems,” *Sante Publique (Paris)*., vol. 28, no. 3, pp. 391–397, 2016.

BIODATA PENULIS



Penulis mempunyai nama lengkap Gusti Bagus Syahrani. Lahir di Banyuwangi pada tanggal 2 Agustus 1996. Penulis merupakan anak pertama dari empat bersaudara. Penulis pernah menempuh pendidikan formal di sekolah dasar SDN 2 Kaligondo, Kabupaten Banyuwangi dari tahun 2002 sampai tahun 2008, kemudian penulis melanjutkan di sekolah negeri SMPN 2 Genteng dari tahun 2008 hingga lulus pada tahun 2011, dan SMAN 1 Genteng mulai tahun 2011 sampai 2014. Setelah lulus sekolah menengah akhir, penulis mengikuti SNMPTN dan diterima di Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama menjadi mahasiswa, penulis aktif dalam kegiatan yang bersifat akademis dan non-akademis. Penulis bergabung dalam organisasi Himpunan Mahasiswa Sistem Informasi (HMSI) pada tahun 2015 sebagai staff Media Informasi. Selanjutnya penulis aktif di HMSI pada tahun 2016/2017 sebagai staff ahli pada Media Informasi Tercatat penulis pernah menjalani kerja praktik di PT Bank Rakyat Indonesia (BRI) Divisi Satelit dalam departemen Teknologi Informasi selama dua bulan pada tahun 2017. Untuk mendapatkan gelar Sarjana Komputer (S.Kom), penulis mengambil laboratorium bidang minat Rekayasa Data dan Intelegensi Bisnis (RDIB) dengan topik tugas akhir pada Optimasi. Untuk kepentingan penelitian penulis juga dapat menghubungi melalui e-mail: gustibs35@gmail.com

LAMPIRAN A : Jadwal UAS Manual

Hari	Jam	Mata Kuliah	Kelas	Ruang
Senin	08.00 - 10.00	Menggambar Teknik	A	IE - 106
		Menggambar Teknik	B	IE - 103
		Menggambar Teknik	C	IE - 108
		Menggambar Teknik	D	IE - 104
		Menggambar Teknik	Q	IE - 109
		Manajemen Persediaan dan Pengadaan	A	IE - 105
		Manajemen Distribusi	A	Lab. MM
		Ergonomi Industri	A	IE - 101
		Ergonomi Industri	B	Auditorium Sinarmas
		Ergonomi Industri	C	
		Ergonomi Industri	D	
		Ergonomi Industri	Q	IE - 102
	10.30 - 12.30	Analisa Produktivitas	A	IE - 106
		Teori Permainan	A	IE - 105
		Pengantar Teknik dan Sistem Industri	A	IE - 104
		Pengantar Teknik dan Sistem Industri	B	IE - 103
		Pengantar Teknik dan Sistem Industri	C	Auditorium Sinarmas
		Pengantar Teknik dan Sistem Industri	D	
		Pengantar Teknik dan Sistem Industri	Q	IE - 109
	13.30 - 15.30	Pengetahuan Bahan Teknik	A	IE - 101
		Pengetahuan Bahan Teknik	B	IE - 102
		Pengetahuan Bahan Teknik	C	IE - 103

Hari	Jam	Mata Kuliah	Kelas	Ruang
Selasa		Pengetahuan Bahan Teknik	D	IE - 104
		Pengetahuan Bahan Teknik	Q	IE - 105
		Analisa Keputusan	A	IE - 108
	08.00 - 10.00	Pemeliharaan dan Teknik Keandalan	A	IE - 109
		Pemeliharaan dan Teknik Keandalan	B	IE - 108
		Pemeliharaan dan Teknik Keandalan	C	Auditorium Sinarmas
		Pemeliharaan dan Teknik Keandalan	Q	
	10.30 - 12.30	Otomasi Industri	A	Lab. MM
		Otomasi Industri	B	Auditorium Sinarmas
		Otomasi Industri	C	
		Otomasi Industri	D	
		Otomasi Industri	Q	IE - 109
		Faal dan Biomenikanika Kerja	A	Lab. E & PSK
		Six Sigma	A	IE - 105
		Metodologi Sistem Dinamik	A	IE - 108
		Perancangan dan Pengembangan Produk	A	IE - 101
		Perancangan dan Pengembangan Produk	B	IE - 102
		Perancangan dan Pengembangan Produk	C	IE - 103
		Perancangan dan Pengembangan Produk	Q	IE - 104
	13.30 - 15.30	Perancangan Fasilitas	A	Auditorium Sinarmas

Hari	Jam	Mata Kuliah	Kelas	Ruang
		Perancangan Fasilitas	B	
		Perancangan Fasilitas	Q	
		Perancangan Fasilitas	C	IE - 109
		Metodologi Penyelesaian Masalah	A	IE - 106
		Manajemen Lingkungan Industri	A	IE - 108
		Rekayasa Proses Bisnis	A	IE - 105
Rabu	08.00 - 10.00	Perencanaan Industri II	A	Auditorium Sinarmas
		Perencanaan Industri II	B	
		Perencanaan Industri II	C	
		Perencanaan Industri II	D	
		Perencanaan Industri II	Q	IE - 109
		Perencanaan Industri II	Z	IE - 109
		Pengantar Ilmu Ekonomi	A	IE - 101
		Pengantar Ilmu Ekonomi	B	IE - 102
		Pengantar Ilmu Ekonomi	C	IE - 103
		Pengantar Ilmu Ekonomi	D	IE - 104
		Pengantar Ilmu Ekonomi	Q	IE - 105
	10.30 - 12.30	Proses Manufaktur	A	IE - 103
		Proses Manufaktur	B	IE - 108
		Proses Manufaktur	Q	IE - 104

Hari	Jam	Mata Kuliah	Kelas	Ruang
Kamis		Penelitian Operasional II	A	Auditorium Sinarmas
		Penelitian Operasional II	B	
		Penelitian Operasional II	C	
		Penelitian Operasional II	Q	IE - 109
		Aplikasi Ergonomi Industri	A	IE - 105
	13.30 - 15.30	Analisis dan Estimasi Biaya	A	Auditorium Sinarmas
		Analisis dan Estimasi Biaya	B	
		Analisis dan Estimasi Biaya	C	IE - 103
		Analisis dan Estimasi Biaya	D	IE - 104
		Analisis dan Estimasi Biaya	Q	IE - 109
		Makro Ergonomi	A	IE - 105
		Manajemen Resiko Korporat	A	IE - 108
		Computer Integrated Manufacturing	A	IE - 102
		Permodelan Sistem Berbasis Agen	A	Lab. MM
		Statistik Industri II	A	IE - 103
		Statistik Industri II	B	Auditorium Sinarmas
		Statistik Industri II	D	
		Statistik Industri II	C	IE - 104
		Statistik Industri II	Q	IE - 109
		Simulasi Diskrit Terapan	A	IE - 105
	08.00 - 10.00			
	10.00 - 12.30	Perencanaan dan Pengendalian Produksi	A	IE - 109

Hari	Jam	Mata Kuliah	Kelas	Ruang
		Perencanaan dan Pengendalian Produksi	B	IE - 105
		Perencanaan dan Pengendalian Produksi	C	Auditorium Sinarmas
		Perencanaan dan Pengendalian Produksi	Q	
	13.30 - 15.30	Teknik Pengendalian Kualitas	A	IE - 109
		Teknik Pengendalian Kualitas	B	Auditorium Sinarmas
		Teknik Pengendalian Kualitas	C	
		Teknik Pengendalian Kualitas	Q	
		Mekanika Teknik	A	IE - 108
		Ergo Safety	A	IE - 105
		Data Mining	A	IE - 106
		Manajemen Pengetahuan	A	Lab. MM
Jum'at	07.00 - 09.00	Perancangan Sistem Informasi Bisnis	A	Auditorium Sinarmas
		Perancangan Sistem Informasi Bisnis	B	
		Perancangan Sistem Informasi Bisnis	C	IE - 105
		Perancangan Sistem Informasi Bisnis	Q	IE - 109
		Supply Chain Management	A	IE - 108
	09.30 - 11.30	Termodinamika	A	IE - 103
		Termodinamika	B	IE - 104
		Termodinamika	C	IE - 105
	13.30 - 15.30	Matematika Optimasi	A	IE - 101

Hari	Jam	Mata Kuliah	Kelas	Ruang
		Matematika Optimasi	B	IE - 102
		Matematika Optimasi	C	IE - 103
		Matematika Optimasi	D	IE - 104
		Matematika Optimasi	Q	IE - 109

LAMPIRAN B : Inisiasi Jadwal UAS

Hari	Jam	Mata Kuliah	Ruang Kelas
Senin	08.00-10.00	Perancangan Sistem Informasi Bisnis	IE - 101
Senin	08.00-10.00	Perancangan Sistem Informasi Bisnis	IE - 102
Senin	08.00-10.00	Perancangan Sistem Informasi Bisnis	IE - 103
Senin	08.00-10.00	Perancangan Sistem Informasi Bisnis	IE - 104
		Proses Manufaktur	
Senin	08.00-10.00	Proses Manufaktur	IE - 105
Senin	08.00-10.00	Proses Manufaktur	IE - 106
Senin	08.00-10.00	Proses Manufaktur	IE - 108
Senin	10.30-12.30	Computer Integrated Manufacturing	IE - 101
		Manajemen Pengetahuan	
		Pengetahuan Bahan Teknik	
Senin	10.30-12.30	Pengetahuan Bahan Teknik	IE - 102
Senin	10.30-12.30	Pengetahuan Bahan Teknik	IE - 103
Senin	10.30-12.30	Pengetahuan Bahan Teknik	IE - 104
Senin	10.30-12.30	Pengetahuan Bahan Teknik	IE - 105
Senin	10.30-12.30	Pengetahuan Bahan Teknik	IE - 106
		Statistik Industri II	
Senin	10.30-12.30	Statistik Industri II	IE - 108

Hari	Jam	Mata Kuliah	Ruang Kelas
Senin	10.30-12.30	Statistik Industri II	IE - 109
Senin	10.30-12.30	Statistik Industri II	Auditorium Sinarmas
Senin	13.30-15.30	Otomasi Industri	IE - 101
Senin	13.30-15.30	Otomasi Industri	IE - 102
Senin	13.30-15.30	Otomasi Industri	IE - 103
Senin	13.30-15.30	Otomasi Industri	IE - 104
Senin	13.30-15.30	Otomasi Industri	IE - 105
		Perencanaan Industri II	
Senin	13.30-15.30	Perencanaan Industri II	IE - 106
Senin	13.30-15.30	Perencanaan Industri II	IE - 108
Senin	13.30-15.30	Perencanaan Industri II	IE - 109
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri	IE - 101
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri	IE - 102
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri	IE - 103
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri	IE - 104
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri	IE - 105
		Perencanaan dan Pengendalian Produksi	
Selasa	08.00-10.00	Perencanaan dan Pengendalian Produksi	IE - 106

Hari	Jam	Mata Kuliah	Ruang Kelas
Selasa	08.00-10.00	Perencanaan dan Pengendalian Produksi	IE - 108
Selasa	08.00-10.00	Perencanaan dan Pengendalian Produksi	IE - 109
		Simulasi Diskrit Terapan	
Selasa	08.00-10.00	Simulasi Diskrit Terapan	Auditorium Sinarmas
Selasa	10.30-12.30	Ergonomi Industri	IE - 101
Selasa	10.30-12.30	Ergonomi Industri	IE - 102
Selasa	10.30-12.30	Ergonomi Industri	IE - 103
Selasa	10.30-12.30	Ergonomi Industri	IE - 104
Selasa	10.30-12.30	Ergonomi Industri	IE - 105
		Menggambar Teknik	
Selasa	10.30-12.30	Menggambar Teknik	IE - 106
Selasa	10.30-12.30	Menggambar Teknik	IE - 108
Selasa	10.30-12.30	Menggambar Teknik	IE - 109
Selasa	10.30-12.30	Menggambar Teknik	Auditorium Sinarmas
		Pemeliharaan dan Teknik Keandalan	
Selasa	13.30-15.30	Faal dan Biomenikanika Kerja	IE - 101
		Manajemen Persediaan dan Pengadaan	
		Matematika Optimasi	
Selasa	13.30-15.30	Matematika Optimasi	IE - 102

Hari	Jam	Mata Kuliah	Ruang Kelas
Selasa	13.30-15.30	Matematika Optimasi	IE - 103
Selasa	13.30-15.30	Matematika Optimasi	IE - 104
Selasa	13.30-15.30	Matematika Optimasi	IE - 105
Selasa	13.30-15.30	Matematika Optimasi	IE - 106
Rabu	08.00-10.00	Penelitian Operasional II	IE - 101
Rabu	08.00-10.00	Penelitian Operasional II	IE - 102
Rabu	08.00-10.00	Penelitian Operasional II	IE - 103
Rabu	08.00-10.00	Penelitian Operasional II	IE - 104
		Pengantar Ilmu Ekonomi	
Rabu	08.00-10.00	Pengantar Ilmu Ekonomi	IE - 105
Rabu	08.00-10.00	Pengantar Ilmu Ekonomi	IE - 106
Rabu	08.00-10.00	Pengantar Ilmu Ekonomi	IE - 108
Rabu	08.00-10.00	Pengantar Ilmu Ekonomi	IE - 109
	Teori Permainan		
Rabu	10.30-12.30	Permodelan Sistem Berbasis Agen	IE - 101
		Supply Chain Management	
		Termodinamika	
Rabu	10.30-12.30	Termodinamika	IE - 102
Rabu	10.30-12.30	Termodinamika	IE - 103

Hari	Jam	Mata Kuliah	Ruang Kelas
Rabu	10.30-12.30	Termodinamika	IE - 104
Rabu	13.30-15.30	Aplikasi Ergonomi Industri	IE - 101
		Manajemen Distribusi	
Rabu	13.30-15.30	Manajemen Distribusi	IE - 102
		Mekanika Teknik	
		Perancangan Fasilitas	
Rabu	13.30-15.30	Perancangan Fasilitas	IE - 103
Rabu	13.30-15.30	Perancangan Fasilitas	IE - 104
Rabu	13.30-15.30	Perancangan Fasilitas	IE - 105
Kamis	08.00-10.00	Analisa Keputusan	IE - 101
		Analisis dan Estimasi Biaya	
Kamis	08.00-10.00	Analisis dan Estimasi Biaya	IE - 102
Kamis	08.00-10.00	Analisis dan Estimasi Biaya	IE - 103
Kamis	08.00-10.00	Analisis dan Estimasi Biaya	IE - 104
Kamis	08.00-10.00	Analisis dan Estimasi Biaya	IE - 105
Kamis	10.30-12.30	Metodologi Sistem Dinamik	IE - 101
		Six Sigma	
		Teknik Pengendalian Kualitas	
Kamis	10.30-12.30	Teknik Pengendalian Kualitas	IE - 102
Kamis	10.30-12.30	Teknik Pengendalian Kualitas	IE - 103

Hari	Jam	Mata Kuliah	Ruang Kelas
Kamis	10.30-12.30	Teknik Pengendalian Kualitas	IE - 104
Kamis	13.30-15.30	Manajemen Lingkungan Industri	IE - 101
		Metodologi Penyelesaian Masalah	
		Perancangan dan Pengembangan Produk	
Kamis	13.30-15.30	Perancangan dan Pengembangan Produk	IE - 102
Kamis	13.30-15.30	Perancangan dan Pengembangan Produk	IE - 103
Kamis	13.30-15.30	Perancangan dan Pengembangan Produk	IE - 104
		Rekayasa Proses Bisnis	
Kamis	13.30-15.30	Rekayasa Proses Bisnis	IE - 105
Jumat	08.00-10.00	Analisa Produktivitas	IE - 101
Jumat	10.30-12.30	Makro Ergonomi	IE - 101
		Manajemen Resiko Korporat	
Jumat	13.30-15.30	Data Mining	IE - 101
		Ergo Safety	
Jumat	13.30-15.30	Ergo Safety	IE - 102

LAMPIRAN C : Hasil Optimasi Jadwal UAS

Hari	Jam	Mata Kuliah	Ruang
Senin	08.00-10.00	Otomasi Industri	IE - 101
		Otomasi Industri	IE - 102
		Otomasi Industri	IE - 103
		Otomasi Industri	IE - 104
		Otomasi Industri	IE - 105
		Perencanaan Industri II	IE - 105
		Perencanaan Industri II	IE - 106
		Perencanaan Industri II	IE - 108
		Perencanaan Industri II	IE - 109
	10.30-12.30	Pengantar Teknik dan Sistem Industri	IE - 101
		Pengantar Teknik dan Sistem Industri	IE - 102
		Pengantar Teknik dan Sistem Industri	IE - 103
		Pengantar Teknik dan Sistem Industri	IE - 104
		Pengantar Teknik dan Sistem Industri	IE - 105
		Six Sigma	IE - 105
		Teknik Pengendalian Kualitas	IE - 105
		Teknik Pengendalian Kualitas	IE - 106
		Teknik Pengendalian Kualitas	IE - 108
		Teknik Pengendalian Kualitas	IE - 109
	13.30-15.30	Analisa Keputusan	IE - 101
		Manajemen Persediaan dan Pengadaan	IE - 101
		Manajemen Persediaan dan Pengadaan	IE - 102
		Termodinamika	IE - 102
		Termodinamika	IE - 103

Hari	Jam	Mata Kuliah	Ruang
		Termodinamika	IE - 104
Selasa	08.00-10.00	Manajemen Lingkungan Industri	IE - 101
		Metodologi Penyelesaian Masalah	IE - 101
		Proses Manufaktur	IE - 101
		Proses Manufaktur	IE - 102
		Proses Manufaktur	IE - 103
		Proses Manufaktur	IE - 104
		Rekayasa Proses Bisnis	IE - 104
	10.30-12.30	Pemeliharaan dan Teknik Keandalan	IE - 101
		Pemeliharaan dan Teknik Keandalan	IE - 102
		Pemeliharaan dan Teknik Keandalan	IE - 103
		Pemeliharaan dan Teknik Keandalan	IE - 104
		Pengantar Ilmu Ekonomi	IE - 104
		Pengantar Ilmu Ekonomi	IE - 105
		Pengantar Ilmu Ekonomi	IE - 106
		Pengantar Ilmu Ekonomi	IE - 108
		Pengantar Ilmu Ekonomi	IE - 109
	13.30-15.30	Metodologi Sistem Dinamik	IE - 101
		Statistik Industri II	IE - 101
		Statistik Industri II	IE - 102
		Statistik Industri II	IE - 103
		Statistik Industri II	IE - 104
		Statistik Industri II	IE - 105
Rabu	08.00-10.00	Aplikasi Ergonomi Industri	IE - 101
		Manajemen Distribusi	IE - 101
		Manajemen Distribusi	IE - 102

Hari	Jam	Mata Kuliah	Ruang
		Perancangan Fasilitas	IE - 102
		Perancangan Fasilitas	IE - 103
		Perancangan Fasilitas	IE - 104
		Perancangan Fasilitas	IE - 105
	10.30-12.30	Mekanika Teknik	IE - 101
		Supply Chain Management	IE - 101
		Supply Chain Management	IE - 102
	13.30-15.30	Perancangan Sistem Informasi Bisnis	IE - 101
		Perancangan Sistem Informasi Bisnis	IE - 102
		Perancangan Sistem Informasi Bisnis	IE - 103
		Perancangan Sistem Informasi Bisnis	IE - 104
Kamis	08.00-10.00	Analisa Produktivitas	IE - 101
		Analisis dan Estimasi Biaya	IE - 101
		Analisis dan Estimasi Biaya	IE - 102
		Analisis dan Estimasi Biaya	IE - 103
		Analisis dan Estimasi Biaya	IE - 104
		Analisis dan Estimasi Biaya	IE - 105
		Teori Permainan	IE - 105
	10.30-12.30	Menggambar Teknik	IE - 101
		Menggambar Teknik	IE - 102
		Menggambar Teknik	IE - 103
		Menggambar Teknik	IE - 104
		Menggambar Teknik	IE - 105
		Perencanaan dan Pengendalian Produksi	IE - 105
		Perencanaan dan Pengendalian Produksi	IE - 106

Hari	Jam	Mata Kuliah	Ruang
		Perencanaan dan Pengendalian Produksi	IE - 108
		Perencanaan dan Pengendalian Produksi	IE - 109
	13.30-15.30	Faal dan Biomenikanika Kerja	IE - 101
		Matematika Optimasi	IE - 101
		Matematika Optimasi	IE - 102
		Matematika Optimasi	IE - 103
		Matematika Optimasi	IE - 104
		Matematika Optimasi	IE - 105
Jumat	08.00-10.00	Computer Integrated Manufacturing	IE - 101
		Makro Ergonomi	IE - 101
		Manajemen Resiko Korporat	IE - 101
		Manajemen Resiko Korporat	IE - 102
		Permodelan Sistem Berbasis Agen	IE - 102
	10.30-12.30	Data Mining	IE - 101
		Ergo Safety	IE - 101
		Ergo Safety	IE - 102
		Manajemen Pengetahuan	IE - 102
		Perancangan dan Pengembangan Produk	IE - 102
		Perancangan dan Pengembangan Produk	IE - 103
		Perancangan dan Pengembangan Produk	IE - 104
		Perancangan dan Pengembangan Produk	IE - 105
	13.30-15.30	Ergonomi Industri	IE - 101
		Ergonomi Industri	IE - 102
		Ergonomi Industri	IE - 103
		Ergonomi Industri	IE - 104

Hari	Jam	Mata Kuliah	Ruang
		Ergonomi Industri	IE - 105
		Penelitian Operasional II	IE - 105
		Penelitian Operasional II	IE - 106
		Penelitian Operasional II	IE - 108
		Penelitian Operasional II	IE - 109
		Pengetahuan Bahan Teknik	IE - 109
		Pengetahuan Bahan Teknik	Auditorium Sinarmas
		Simulasi Diskrit Terapan	Auditorium Sinarmas

LAMPIRAN D : Visualisasi Inisiasi Jadwal UAS

Hari	Jam	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas	Lab. MM	IE - 303 D	IE - 302
Senin	08.00-10.00	Perancangan Sistem Informasi Bisnis (1 - 45)	Perancangan Sistem Informasi Bisnis (46 - 90)	Perancangan Sistem Informasi Bisnis (91 - 135)	Perancangan Sistem Informasi Bisnis (136 - 160) Proses Manufaktur (1 - 20)	Proses Manufaktur (21 - 65)	Proses Manufaktur (66 - 105)	Proses Manufaktur (106 - 107)					
Senin	10.30-12.30	Computer Integrated Manufacturing (1 - 10) Manajemen Pengetahuan (1 - 23) Pengetahuan Bahan Teknik (1 - 12)	Pengetahuan Bahan Teknik (13 - 57)	Pengetahuan Bahan Teknik (58 - 102)	Pengetahuan Bahan Teknik (103 - 147)	Pengetahuan Bahan Teknik (148 - 192)	Pengetahuan Bahan Teknik (193 - 194) Statistik Industri II (1 - 38)	Statistik Industri II (39 - 78)	Statistik Industri II (79 - 148)	Statistik Industri II (149 - 199)			
Senin	13.30-15.30	Otomasi Industri (1 - 45)	Otomasi Industri (46 - 90)	Otomasi Industri (91 - 135)	Otomasi Industri (136 - 180)	Otomasi Industri (181 - 189) Perencanaan Industri II (1 - 36)	Perencanaan Industri II (37 - 76)	Perencanaan Industri II (77 - 116)	Perencanaan Industri II (117 - 147)				
Selasa	08.00-10.00	Pengantar Teknik dan Sistem Industri (1 - 45)	Pengantar Teknik dan Sistem Industri (46 - 90)	Pengantar Teknik dan Sistem Industri (91 - 135)	Pengantar Teknik dan Sistem Industri (136 - 180)	Pengantar Teknik dan Sistem Industri (181 - 202) Perencanaan dan Pengendalian Produksi (1 - 23)	Perencanaan dan Pengendalian Produksi (24 - 63)	Perencanaan dan Pengendalian Produksi (64 - 103)	Perencanaan dan Pengendalian Produksi (104 - 170)	Simulasi Diskrit Terapan (171 - 184)			
Selasa	10.30-12.30	Ergonomi Industri (1 - 45)	Ergonomi Industri (46 - 90)	Ergonomi Industri (91 - 135)	Ergonomi Industri (136 - 180)	Ergonomi Industri (181 - 183) Menggambar Teknik (1 - 42)	Menggambar Teknik (43 - 82)	Menggambar Teknik (83 - 122)	Menggambar Teknik (123 - 192)	Menggambar Teknik (193 - 194) Pemeliharaan dan Teknik Keandalan (1 - 152)			
Selasa	13.30-15.30	Faal dan Biomekanika Kerja (1 - 7) Manajemen Persediaan dan Pengadaan (1 - 22) Matematika Optimasi (1 - 16)	Matematika Optimasi (17 - 61)	Matematika Optimasi (62 - 106)	Matematika Optimasi (107 - 151)	Matematika Optimasi (152 - 196)	Matematika Optimasi (197 - 207)						
Rabu	08.00-10.00	Penelitian Operasional II (1 - 45)	Penelitian Operasional II (46 - 90)	Penelitian Operasional II (91 - 135)	Penelitian Operasional II (136 - 144) Pengantar Ilmu Ekonomi (1 - 36)	Pengantar Ilmu Ekonomi (37 - 81)	Pengantar Ilmu Ekonomi (82 - 121)	Pengantar Ilmu Ekonomi (122 - 161)	Pengantar Ilmu Ekonomi (162 - 200) Teori Permainan (1 - 16)				
Rabu	10.30-12.30	Permodelan Sistem Berbasis Agen (1 - 18) Supply Chain Management (1 - 25)	Termodinamika (26 - 70)	Termodinamika (71 - 115)	Termodinamika (116 - 119)								
Rabu	13.30-15.30	Aplikasi Ergonomi Industri (1 - 22) Manajemen Distribusi (1 - 23)	Manajemen Distribusi (24 - 24) Mekanika Teknik (1 - 25) Perancangan Fasilitas (1 - 19)	Perancangan Fasilitas (20 - 64)	Perancangan Fasilitas (65 - 109)	Perancangan Fasilitas (110 - 142)							

LAMPIRAN E : Visualisasi Hasil Optimasi Jadwal UAS

Hari	Jam	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas	Lab. MM	IE - 303 D	IE - 302
Senin	08.00- 10.00	Otomasi Industri (1 - 45)	Otomasi Industri (46 - 90)	Otomasi Industri (91 - 135)	Otomasi Industri (136 - 180)	Otomasi Industri (181 - 189) Perencanaan Industri II (1 - 36)	Perencanaan Industri II (37 - 76)	Perencanaan Industri II (77 - 116)	Perencanaan Industri II (117 - 147)				
Senin	10.30- 12.30	Pengantar Teknik dan Sistem Industri (1 - 45)	Pengantar Teknik dan Sistem Industri (46 - 90)	Pengantar Teknik dan Sistem Industri (91 - 135)	Pengantar Teknik dan Sistem Industri (136 - 180)	Pengantar Teknik dan Sistem Industri (181 - 202) Six Sigma (1 - 22)	Teknik Pengendalian Kualitas (23 - 62)	Teknik Pengendalian Kualitas (63 - 102)	Teknik Pengendalian Kualitas (103 - 163)				
Senin	13.30- 15.30	Analisa Keputusan (1 - 24) Manajemen Persediaan dan Pengadaan (1 - 21)	Manajemen Persediaan dan Pengadaan (22 - 22) Termodinamika (1 - 44)	Termodinamika (45 - 89)	Termodinamika (90 - 94)								
Selasa	08.00- 10.00	Manajemen Lingkungan Industri (1 - 22) Metodologi Penyelesaian Masalah (1 - 23)	Proses Manufaktur (24 - 68)	Proses Manufaktur (69 - 113)	Proses Manufaktur (114 - 130) Rekayasa Proses Bisnis (1 - 22)								
Selasa	10.30- 12.30	Pemeliharaan dan Teknik Keandalan (1 - 45)	Pemeliharaan dan Teknik Keandalan (46 - 90)	Pemeliharaan dan Teknik Keandalan (91 - 135)	Pemeliharaan dan Teknik Keandalan (136 - 152) Pengantar Ilmu Ekonomi (1 - 28)	Pengantar Ilmu Ekonomi (29 - 73)	Pengantar Ilmu Ekonomi (74 - 113)	Pengantar Ilmu Ekonomi (114 - 153)	Pengantar Ilmu Ekonomi (154 - 200)				
Selasa	13.30- 15.30	Metodologi Sistem Dinamik (1 - 12) Statistik Industri II (1 - 33)	Statistik Industri II (34 - 78)	Statistik Industri II (79 - 123)	Statistik Industri II (124 - 168)	Statistik Industri II (169 - 199)							
Rabu	08.00- 10.00	Aplikasi Ergonomi Industri (1 - 22) Manajemen Distribusi (1 - 23)	Manajemen Distribusi (24 - 24) Perancangan Fasilitas (1 - 44)	Perancangan Fasilitas (45 - 89)	Perancangan Fasilitas (90 - 134)	Perancangan Fasilitas (135 - 142)							
Rabu	10.30- 12.30	Mekanika Teknik (1 - 25) Supply Chain Management (1 - 20)	Supply Chain Management (21 - 25)										
Rabu	13.30- 15.30	Perancangan Sistem Informasi Bisnis (1 - 45)	Perancangan Sistem Informasi Bisnis (46 - 90)	Perancangan Sistem Informasi Bisnis (91 - 135)	Perancangan Sistem Informasi Bisnis (136 - 160)								

Hari	Jam	IE - 101	IE - 102	IE - 103	IE - 104	IE - 105	IE - 106	IE - 108	IE - 109	Auditorium Sinarmas	Lab. MM	IE - 303 D	IE - 302
Kamis	10.30-12.30	Menggambar Teknik (1 - 45)	Menggambar Teknik (46 - 90)	Menggambar Teknik (91 - 135)	Menggambar Teknik (136 - 180)	Menggambar Teknik (181 - 194) Perencanaan dan Pengendalian Produksi (1 - 31)	Perencanaan dan Pengendalian Produksi (32 - 71)	Perencanaan dan Pengendalian Produksi (72 - 111)	Perencanaan dan Pengendalian Produksi (112 - 170)				
Kamis	13.30-15.30	Faal dan Biomekanika Kerja (1 - 7) Matematika Optimasi (1 - 38)	Matematika Optimasi (39 - 83)	Matematika Optimasi (84 - 128)	Matematika Optimasi (129 - 173)	Matematika Optimasi (174 - 207)							
Jumat	08.00-10.00	Computer Integrated Manufacturing (1 - 10) Makro Ergonomi (1 - 20) Manajemen Resiko Korporat (1 - 15)	Manajemen Resiko Korporat (16 - 23) Permodelan Sistem Berbasis Agen (1 - 18)										
Jumat	08.00-10.00	Computer Integrated Manufacturing (1 - 10) Makro Ergonomi (1 - 20) Manajemen Resiko Korporat (1 - 15)	Manajemen Resiko Korporat (16 - 23) Permodelan Sistem Berbasis Agen (1 - 18)										
Jumat	10.30-12.30	Data Mining (1 - 22) Ergo Safety (1 - 23)	Ergo Safety (24 - 24) Manajemen Pengetahuan (1 - 23) Perancangan dan Pengembangan Produk (1 - 21)	Perancangan dan Pengembangan Produk (22 - 66)	Perancangan dan Pengembangan Produk (67 - 111)	Perancangan dan Pengembangan Produk (112 - 134)							
Jumat	13.30-15.30	Ergonomi Industri (1 - 45)	Ergonomi Industri (46 - 90)	Ergonomi Industri (91 - 135)	Ergonomi Industri (136 - 180)	Ergonomi Industri (181 - 183) Penelitian Operasional II (1 - 42)	Penelitian Operasional II (43 - 82)	Penelitian Operasional II (83 - 122)	Penelitian Operasional II (123 - 144) Pengetahuan Bahan Teknik (1 - 48)	Pengetahuan Bahan Teknik (49 - 194) Simulasi Diskrit Terapan (1 - 14)			